Document Number: P2525R0
Date: 2022-01-15
Authors: Michael Wong
Project: Programming Language C++, SG19 Machine Learning
Reply to: Michael Wong <michael@codeplay.com>

SG19: Machine Learning Meeting Minutes 2020/12/10-2022/01/13

Contents

Minutes for 2020/12/10 SG19 Conference Call

On Wed, Dec 9, 2020 at 11:12 AM Michael Wong <fraggamuffin_at_[hidden]> wrote:
> *SG19 Machine Learning 2 hours. This session will focus on the 3rd review*
> *of Stats paper but with updates from all the others optionally. Links to*
> *paper coming.*
>
>
> *Hi,*
>
> *Michael Wong is inviting you to a scheduled Zoom meeting.*
>
> *Topic: SG19 monthly Dec 2020-Feb 2021*
> *Time: Dec 10, 2020 02:00 PM Eastern Time (US and Canada)*
> *Every month on the Second Thu, until Feb 11, 2021, 3 occurrence(s)*
> *Dec 10, 2020 02:00 PM ET 1900 UTC*
> *Jan 14, 2021 02:00 PM ET 1900 UTC*
> *Feb 11, 2021 02:00 PM ET 1900 UTC*
> *Please download and import the following iCalendar (.ics) files to*
> *your calendar system.*
> *Monthly:*
> *https://iso.zoom.us/meeting/tJctf-2tpzotGNHL5pZqwtjELee0mcG2zzCi/ics?icsToken=98tyKuCrrjMuH92UtxuCRowqAoqgLO_xmH5ajY11sEr1OTFEdgnTGudHYr98N4rK*
>
> *Join from PC, Mac, Linux, iOS or Android:*
> *https://iso.zoom.us/j/93084591725?pwd=K3QxZjJlcnljaE13ZWU5cTlLNkx0Zz09*
> *Password: 035530*
>
> *Or iPhone one-tap :*
> *US: +13017158592,,93084591725# or +13126266799,,93084591725#*
> *Or Telephone:*
> *Dial(for higher quality, dial a number based on your current*
> *location):*
> *US: +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799 or +1*
> *408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782*
> *or 877 853 5247 (Toll Free)*
> *Meeting ID: 930 8459 1725*
> *Password: 035530*
> *International numbers available: https://iso.zoom.us/u/agewu4X97*
>
> *Or Skype for Business (Lync):*
> *https://iso.zoom.us/skype/93084591725*

>
> *Agenda:*
>
> *1. Opening and introductions*
>
> *The ISO Code of conduct:*
> *https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.pdf*
> *The IEC Code of Conduct:*
>
> *https://basecamp.iec.ch/download/iec-code-of-conduct-for-delegates-and-experts/*
>
> *ISO patent policy.*
>
> *https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Policy.htm?nodeid=6344764&vernum=-2*
>
>
>
> *The WG21 Practices and Procedures and Code of Conduct:*
>
> *https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures*
>
> *1.1 Roll call of participants*
>

Antony Peacock
Joe Sachs
Ozan Irsoy
Phil Ratzloff
Richard Dosselmann
Michael Chiu
Will Wray
William Moses
Scott McMillan
Larry Lewis
Michael Wong
Kens Maurer
Andrew Lumsdaine
Vissil Vassilev
Cyril Khazan

> *1.2 Adopt agenda*
>
Stats paper
combinatorics

> *1.3 Approve minutes from previous meeting, and approve publishing*
> *previously approved minutes to ISOCPP.org*
>
> *1.4 Action items from previous meetings*
>
> *2. Main issues (125 min)*
>
> *2.1 General logistics*
>
> *Meeting plan, focus on one paper per meeting but does not preclude other*
> *paper updates:*
>
> *Dec 10, 2020 02:00 PM ET1900 UTC Stats*
> *Jan 14, 2021 02:00 PM ET 1900 UTCReinforcement Learning and Diff*
> *Calculus*
> *Feb 11, 2021 02:00 PM 1900 UTC ET Graph*
>
> *ISO meeting status*
>
> *future C++ Std meetings*
>
> *2.2 Paper reviews*
>
> *2.2.1: ML topics*
>
> *2.2.1.1 Stats review Richard Dosselman et al*
>
> *P1708R3: Math proposal for Machine Learning: 3rd review*
>
make it a D paper
updated in Sept
SaS mentioned member deviations
 everything in 4.1 are free standing functions
and also most in 4.2 as accumulator design to be done in a 1 pass
5 has discussion and issues

1. all very big in machine learning, competitive with python and sypy
3.1 simple arithmetic mean here, added geometric and harmonic, python did not have weighted versions
3.2 Quantile is asked by Jolanta, in addition to mode using nth largest element
3.3 mode is still there
3.4 skewness are also there and only in numpy extended library and not in python showing how lopsided we are
3.4 kurtosis
3.6 Variance
3.7 Std dev
added walter Brown and Jolanta's concerned items
4. Walter ask that we use same style as <random> so it gets its own header
4.1 using ranges and Eric Niebler's proposal to have less processing on user side, and more things automated
not too many concepts, just enough to protect everything
if you are doing the mode, then we want to able to support text
weighted version can support arithmetic type

4.1.1 Mean
using projections of a range
Also use auto as a return type per Walter Brown
4.1.1 if you are already using a floating value for range, that is automatically your return type, per Jolanta suggestion
projections on the actual values
overloaded versions are still there per Michael

function declaration is close to what std spec synopsis
but what is auto doing here instead of result type, need Walter to answer,
auto without -> at the end meand implementation determines the return type,
unless you show the implementation inline, which is not recommended
the only non-deducible parameter is the result, then the most natural thing
is to do geometric mean of double because everything else is deduced
also had a discussion on the parameter order - yes need that order for
result of template argument and iresult depends on R and P
do you anticipate someone override the result template parameter for
intermediate accumulation as an intended use case? Yes ,
result is the intermediate thing
Ask this as a direct question for Walter and Jens, likely he meant there
should be auto with an arrow on the right end

is there an explicit desire for handling for a custom type for a big float? Why, if not then why have an accumulator type ...? Yes this ties in with Jens why not return the big integer and return the big float on the outside might want everything in geometric mean to have a rational type as a result of doing a division? May be over the top but I can see it

current support for non builtin because result needs to be nonarithmetic type, can you do square root, logs, signed division, this proposal shoudl not be bogged down - just be explicit that this does not support user defined type

not worry about complex

4.1.1
Harmonic
projection compiles ok
4.1.2
quantile
 compute in one pass
using std:: stats::stats_range should be stats::range

example us structure binding and return 2 values

can you pass initializer list instead of q? yes, may be show an example of it as it is a constant

which template parameter to deduce and which to override; so eecution policy, r, C1, P1, can be deduced , but Key, Hash, KeyEqual can be overridden
dont want to give all the over parameters when I just want to customize my hash function, so may be better to just ask for a hash table type

4.2 accumulator object based on boost
allows one list, one pass over the list, computing 3 means in one pass

almost all of these can be specific instantiation of generic deduction ? We discussed this last year, that everything is not far from std:;accumulate/reduce as all a mean is accumulate, so stats people dont understand the details and they prefer a wrapper

a way to minimize rounding error, are the mechanism to specify ways to minimize it? pass a different type for the temporary accumulator else no

way to sort the value by magnitude before adding them - you are also
accumulating the error term, kahan summation ?OK perfect

if goal is to make it easy for non-expert and allow them to choose other
matters ? yes may be its time to address that
we dont have a concept in C++ for arithmetic type, so if we do something
like Kahan SummationHN? it would be hard so document and dont do it

4.4 mode
C and P are less important then others, except classes may be we want
deduction guides later
accumulating object is unique to us
destructor should be constexpr
constexpr on constructor may not be correct as it allocates memory if we
are building a hash table which I am not sure is made constexpr
so reconsider the constexpr here

use common name for all the mean_ variants
for auto can be a range based for loop

5.4 added trimmed mean from jolanta

Internalize customization

Move that we send this paper to LEWG modulo changes from this discussion
6/0/0/0/0
Aim for Jan 15

ship vehicle will be the IS and not a TR because this is independent
Have an implementation

PXXXX: combinatorics: 1st Review
>
> > std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2
> > above is the stats paper that was reviewed in Prague
> > http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19
> >

> > *Review Jolanta Polish feedback.*
> > *http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html*
>
> *Unit library cppcon 2020:*
>
> *https://www.youtube.com/watch?v=aN6-Kz0HqRw&feature=emb_logo*
>
> *2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva*
>
> *Reinforcement Learning proposal:*
>
move to Feb
is there a paper on NN
mdarray using storage could work

2.2.1.3 Graph Proposal Phil Ratsloff et al
>
> *P1709R1: Graph Proposal for Machine Learning*
>
> *P1709R3:*
>
> *https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing*
>
>
> *https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing*
>
> *<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>*
>
 working on concepts for graph and reconcile the proposals

> *2.2.1.4: Differentiable Programming by Marco Foco*
>
> *<*
>
> *https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0ljdUAtSQ/edit*
> >

\>

tensorflow community
Automatic diff as first class citizen of llvm IR : made a plugin that does
AD
Vasil will give rough update of people's desires

\>
\> *2.2.3 any other proposal for reviews?*
\>
\> *2.3 Other Papers and proposals*
\>
\> *P1416R1: SG19 - Linear Algebra for Data Science and Machine Learning*
\>
\> *https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk7xKikK4Yp8/edit#heading=h.tj9hitg7dbtr*
\>
\> *P1415: Machine Learning Layered list*
\>
\> *https://docs.google.com/document/d/1eINFdIXWoetbxjO1OKoI_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr*
\>
\> *2.2.2 SG14 Linear Algebra progress:*
\> *Different layers of proposal*
\>
\> *https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXIK_d0IjdUAtSQ/edit*
\>
\> *2.5 Future F2F meetings:*
\>
\> *2.6 future C++ Standard meetings:*
\> *https://isocpp.org/std/meetings-and-participation/upcoming-meetings*
\>
\> *None*
\>
\> *3. Any other business*
\>
\> *New reflector*
\>
\> *http://lists.isocpp.org/mailman/listinfo.cgi/sg19*
\>

> *Old Reflector*
> *https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19*
> *<https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/sg14>*
>
> *Code and proposal Staging area*
>
> *4. Review*
>
> *4.1 Review and approve resolutions and issues [e.g., changes to SG's*
> *working draft]*
>
> *4.2 Review action items (5 min)*
>
> *5. Closing process*
>
> *5.1 Establish next agenda*
>
> *TBD*
>
> *5.2 Future meeting*
>
>
> *Jan 14, 2021 02:00 PM ET 1900 UTCReinforcement Learning and Diff*
> *Calculus*
> *Feb 11, 2021 02:00 PM 1900 UTC ET Graph*

Minutes for 2021/01/14 SG19 Conference Call

ng from our math library
had identified the min P1415 overview document
applications beyond SG19
need large numerical type for factorials
python has builtin wide integer type

unbounded template T for a numeric type, we have failed to specify for the
complex of T type for float and long double

what operations are declared on that type, can't just say plug in any T

these could also be in C without the templates, may also consider how to do
it in C

2.2.3 any other proposal for reviews?
>
> *2.3 Other Papers and proposals*
>
> *P1416R1: SG19 - Linear Algebra for Data Science and Machine Learning*
>
> *https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk7xKikK4Yp8/edit#heading=h.tj9hitg7dbtr*
>
> *P1415: Machine Learning Layered list*
>
> *https://docs.google.com/document/d/1elNFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr*
>
> *2.2.2 SG14 Linear Algebra progress:*
> *Different layers of proposal*
>
> *https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXIK_d0IjdUAtSQ/edit*
>
> *2.5 Future F2F meetings:*
>
> *2.6 future C++ Standard meetings:*
> *https://isocpp.org/std/meetings-and-participation/upcoming-meetings*
>
> *None*
>
> *3. Any other business*
>
> *New reflector*
>
> *http://lists.isocpp.org/mailman/listinfo.cgi/sg19*
>

> *Old Reflector*
> *https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19*
> *<https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/sg14>*
> *
> *Code and proposal Staging area*
> *
> *4. Review*
> *
> *4.1 Review and approve resolutions and issues [e.g., changes to SG's*
> *working draft]*
> *
> *4.2 Review action items (5 min)*
> *
> *5. Closing process*
> *
> *5.1 Establish next agenda*
> *
> *TBD*
> *
> *5.2 Future meeting*
> *
> *
> *Feb 11, 2021 02:00 PM 1900 UTC ET Graph paper*

Minutes for 2021/02/11 SG19 Conference Call

On Wed, Feb 10, 2021 at 12:53 PM Michael Wong <fraggamuffin_at_[hidden]>
wrote:

> *SG19 Machine Learning 2 hours. This session will focus on Graph but with*
> *updates from all the others*
> *optionally.*
>
> *Link to Automatic differentiation proposal:*
>
>
> *https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-*
> *_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel*
>
> *Hi,*
>
> *Michael Wong is inviting you to a scheduled Zoom meeting.*
>
> *Topic: SG19 monthly Dec 2020-Feb 2021*
> *Time: Jan 14, 2020 02:00 PM Eastern Time (US and Canada)*
> *Every month on the Second Thu, until Feb 11, 2021, 3 occurrence(s)*
> *Dec 10, 2020 02:00 PM ET 1900 UTC DONE*
> *Jan 14, 2021 02:00 PM ET 1900 UTC DONE*
> *Feb 11, 2021 02:00 PM ET 1900 UTC*
> *Please download and import the following iCalendar (.ics) files to your*
> *calendar system.*
> *Monthly:*
>
> *https://iso.zoom.us/meeting/tJctf-*
> *2tpzotGNHL5pZqwtjELee0mcG2zzCi/ics?icsToken=98tyKuCrrjMuH92UtxuCRowqAo*
> *qgLO_xmH5ajY11sEr1OTFEdgnTGudHYr98N4rK*
>
> *Join from PC, Mac, Linux, iOS or Android:*
> *https://iso.zoom.us/j/93084591725?pwd=K3QxZjJlcnIjaE13ZWU5cTlLNkx0Zz09*
> *Password: 035530*
>
> *Or iPhone one-tap :*
> *US: +13017158592,,93084591725# or +13126266799,,93084591725#*

> Or Telephone:
> Dial(for higher quality, dial a number based on your current location):
> US: +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799 or +1
> 408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782
> or 877 853 5247 (Toll Free)
> Meeting ID: 930 8459 1725
> Password: 035530
> International numbers available: https://iso.zoom.us/u/agewu4X97
>
> Or Skype for Business (Lync):
> https://iso.zoom.us/skype/93084591725
>
> Agenda:
>
> 1. Opening and introductions
>
> The ISO Code of conduct:
> https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.pdf
> The IEC Code of Conduct:
>
> https://basecamp.iec.ch/download/iec-code-of-conduct-for-delegates-and-experts/
>
> ISO patent policy.
>
> https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Policy.htm?nodeid=6344764&vernum=-2
>
> The WG21 Practices and Procedures and Code of Conduct:
>
> https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures
>
> 1.1 Roll call of participants
>
> 1.2 Adopt agenda
>
> 1.3 Approve minutes from previous meeting, and approve publishing
> previously approved minutes to ISOCPP.org
>
> 1.4 Action items from previous meetings
>
> 2. Main issues (125 min)

>
> *2.1 General logistics*
>
> *Meeting plan, focus on one paper per meeting but does not preclude other*
> *paper updates:*
>
> *Dec 10, 2020 02:00 PM ET1900 UTC Stats DONE*
> *Jan 14, 2021 02:00 PM ET 1900 UTCReinforcement Learning and Diff*
> *Calculus DONE*
> *Feb 11, 2021 02:00 PM 1900 UTC ET Graph*
>
> *ISO meeting status*
>
> *future C++ Std meetings*
>
> *2.2 Paper reviews*
>
> *2.2.1: ML topics*
>
> *2.2.1.1 Graph Proposal Phil Ratsloff et al*
>
> *P1709R1: Graph Proposal for Machine Learning*
>
> *P1709R3:*
>
> *https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing*
>
>
> *https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing*
>
> *<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>*
>
>
>
<https://docs.google.com/document/d/175wlm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel>
>
D1709R3
input from Andrew sutton/Lumsdaine

embracing ranges

using references for vertices and edges ok for contiguous space but bad for
Deque; these are iterators

east const is now west const to be std consistent

std graph namespace is std so no separate namespace

data structure into their own paper O1-O7

O5 algorithm

connected component

incidence graph a collection of vertex range and vertex edge range

requires undirected graph; very few algorithms requires directed or
undirected

will not have to describe in text what this means

all the other algorithms will be affected in similar way

vertex range and vertex edge range just needs to be forward ranges and this
opens it up for other graphs to use these concepts

directed and undirected is a data property; so 2 overloads for directed
and undirected

this gives one starting point, and no an arbitrary starting point

why graph is mutable ref vs const ref? yes probably cosnst ref

names like vertex are at the top level and may intersect with other names

02- uniform graph interface

traits are here

graph has its own value type and allocator

then the outward and inward range

can I remove the const version of these?

then I can shrink the number of function down

with that we can have concepts

adjacency edge is adjacent vertex through the edges

when algorithm dont use them then refactor them; if the do use them, then
add them to the interface

edge_list graph

concepts behaviorally defined instead of syntactically, can we replace

traits class with customization points
for users who want to bring their own graphs to the library
allow concepts that are type drive and not syntactically driven
like edge_key for their type then they dont need traits

directed graph is an ordered pair; undirected graph is an unordered pair
vertex path is a range of vertex iterators

iterator with vertex type and edge type

added graph_T and graph allocator type

vertex_difference_t

concept I added helps me to consolidate things, else would need to add it
for every different kind of iterators

non_const is better to enable to cover both cases, WW will send you some
patterns; no traits class, just pure deductions

03- Directed Graph interface
is following the template that 02 was building

04- is the adaptors that takes an incidence graph
vertex-vertex range and edge_list_range adaptors

shortest path: 2 paths of that length, do we output one or both?
if they want to bring their own graphs into this it will be an overload

07-undirected_adjacency_list
can talk about this at a higher level

with vertexes, is that a sized range? what is expectation for edge ranges?
in some algorithm you want the degree of each vertex which is the size of
the range
dont want to make requirement on concepts that the algorithm is going to
need, but now the interface will constrain it
some algorithm had both edge, and vertex, but some also had adjacency
pulled vertex list graph as part of definition

CORE list:
P3?
COncepts and algorithms, and no API (Concept is defined in API): 5 2 1 3 0
needs some reorganization

Uniform API, Algorithm and 2 data structues directed 0 5 4 3 0
  Uniform API, Algorithm and 2 data structues directed 0 0 3 7 0

uniform API, algo, and directed adjacency_vector data structure

set of algo, and subset concept requried by those algo; no need for data
structire, if they CPOs then the ycan take as graph; graph withotu algo is
not useful

RFRM has his own paper similar to this graph
your paper is graph only, worry that it woudl be difficult to rewrite my
paper so I can use algo from your paper
N3700 Hierarchical Data structure
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3700.html
how to adapt this data structure as a graph?
there is no graph, should be able cal DFS BFS on vector and not do anything
fancy to that
and people should be able to use graph in some larger context

a concept that has an edge key, and call edge key function, so what type
does it have and what does it return? auto will just return what it return,
excessive typing is old fashion
use decltype to extract actual type
as long as that type satisfies the concept then algorithm should still work
in concept I have an edge key function, then I am not worry the type it
returns

Since I mentioned reddit. Here's the thread I referred to..
https://www.reddit.com/r/cpp/comments/lfc96p/n3700_hierarchical_data_structures_and_related/

> *2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva*
>
> *Reinforcement Learning proposal:*
>
> *2.2.1.3 Differential Calculs:*
>
>
> *https://docs.google.com/document/d/175wlm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel*
>
>
> *2.2.1.4: Stats paper*
>
> *https://github.com/cplusplus/papers/issues/979*
>
> *Stats review Richard Dosselman et al*
>
> *P1708R3: Math proposal for Machine Learning: 3rd review*
>
> *PXXXX: combinatorics: 1st Review*
>
> > *std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2*
> > *above is the stats paper that was reviewed in Prague*
> > *http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19*
> >
> > *Review Jolanta Polish feedback.*
> > *http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html*
>
> *2.2.3 any other proposal for reviews?*

>
> *2.3 Other Papers and proposals*
>
> *P1416R1: SG19 - Linear Algebra for Data Science and Machine Learning*
>
> *https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk7xKikK4Yp8/edit#heading=h.tj9hitg7dbtr*
>
> *P1415: Machine Learning Layered list*
>
> *https://docs.google.com/document/d/1eINFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr*
>
> *2.2.2 SG14 Linear Algebra progress:*
> *Different layers of proposal*
>
> *https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0IjdUAtSQ/edit*
>
> *2.5 Future F2F meetings:*
>
> *2.6 future C++ Standard meetings:*
> *https://isocpp.org/std/meetings-and-participation/upcoming-meetings*
>
> *None*
>
> *3. Any other business*
>
> *New reflector*
>
> *http://lists.isocpp.org/mailman/listinfo.cgi/sg19*
>
> *Old Reflector*
> *https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19*
> *<https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/sg14>*
>
> *Code and proposal Staging area*
>
> *4. Review*
>
> *4.1 Review and approve resolutions and issues [e.g., changes to SG's*

> working draft]
>
> 4.2 Review action items (5 min)
>
> 5. Closing process
>
> 5.1 Establish next agenda
>
> TBD
>
> 5.2 Future meeting
>
> April 8 2021 02:00 PM Eastern Time (1800 UTC): Stats

Minutes for 2021/04/08 SG19 Conference Call

Minutes

On Wed, Apr 7, 2021 at 10:21 AM Michael Wong <fraggamuffin_at_[hidden]> wrote:

> SG19 Machine Learning 2 hours. This session will focus on Stats and
> Combinatorics but with updates from all the others optionally.
>
>
> Hi,
>
> Michael Wong is inviting you to a scheduled Zoom meeting.
>
> Topic: SG19 monthly Dec 2020-Feb 2021
> Time: Apr 8, 2020 02:00 PM Eastern Time (US and Canada) Stats
> Every month on the Second Thu,
> May 13, 2020 02:00 PM ET 1900 UTC Reinformaent Learning and Diff
> Calculus
> June 10, 2021 02:00 PM ET 1900 UTC Graph
> Jul 8, 2021 02:00 PM ET 1900 UTC Stats and Combinatorics
> Please download and import the following iCalendar (.ics) files to your
> calendar system.
> Monthly:
>
> https://iso.zoom.us/meeting/tJctf-
2tpzotGNHL5pZqwtjELee0mcG2zzCi/ics?icsToken=98tyKuCrrjMuH92UtxuCRowqAo
qgLO_xmH5ajY11sEr1OTFEdgnTGudHYr98N4rK
>
> Join from PC, Mac, Linux, iOS or Android:
> https://iso.zoom.us/j/93084591725?pwd=K3QxZjJlcnljaE13ZWU5cTlLNkx0Zz09
> Password: 035530
>
> Or iPhone one-tap :
> US: +13017158592,,93084591725# or +13126266799,,93084591725#
> Or Telephone:
> Dial(for higher quality, dial a number based on your current location):
> US: +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799 or +1
> 408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782
> or 877 853 5247 (Toll Free)
> Meeting ID: 930 8459 1725
> Password: 035530
> International numbers available: https://iso.zoom.us/u/agewu4X97
>

> *Or Skype for Business (Lync):*
> *https://iso.zoom.us/skype/93084591725*
> *
> *Agenda:*
> *
> *1. Opening and introductions*
> *
> *The ISO Code of conduct:*
> *https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.pdf*
> *The IEC Code of Conduct:*
> *
> *https://basecamp.iec.ch/download/iec-code-of-conduct-for-delegates-and-experts/*
> *
> *ISO patent policy.*
> *
> *https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Policy.htm?nodeid=6344764&vernum=-2*
> *
> *The WG21 Practices and Procedures and Code of Conduct:*
> *
> *https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures*
> *
> *1.1 Roll call of participants*
> *

Andrew Lumsdaine, Guy Davidson, Johan Lundberg, Ozran Irsoy, Phil Ratzloff, Scott McMillan, Scott Moe, Will Wray, Michael Wong , Jens Maurer, Rene Rivera, Cyril Khazan,

1.2 Adopt agenda
> *
> *1.3 Approve minutes from previous meeting, and approve publishing*
> *previously approved minutes to ISOCPP.org*
> *
> *1.4 Action items from previous meetings*
> *
> *2. Main issues (125 min)*
> *
> *2.1 General logistics*
> *
> *Meeting plan, focus on one paper per meeting but does not preclude other*
> *paper updates:*

>

> *May 13, 2020 02:00 PM ET 1900 UTC Reinformaent Learning and Diff*
> *Calculus*
> *June 10, 2021 02:00 PM ET 1900 UTC Graph*
> *Jul 8, 2021 02:00 PM ET 1900 UTC Stats and Combinatorics*
>

> *ISO meeting status*
>

> *future C++ Std meetings*
>

> *2.2 Paper reviews*
>

> *2.2.1: ML topics*
>

> *2.2.1.1 Graph Proposal Phil Ratsloff et al*
>

> *P1709R1: Graph Proposal for Machine Learning*
>

> *P1709R3:*
>

> *https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing*
>

>

> *https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing*
>

> *<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>*
>

> *<*
> *https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel*
> >
>

customization point with multiple parameters, how to do that
Eric Niebler for help?

2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva
>

> *Reinforcement Learning proposal:*
>
> *2.2.1.3 Differential Calculus:*
>
>
> *https://docs.google.com/document/d/175wlm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel*
>
>
> *2.2.1.4: Stats paper*
>
> *Current github*
>
> *https://github.com/cplusplus/papers/issues/475*
>
> *https://github.com/cplusplus/papers/issues/979*
>

Review by Johan:
> *I have four comments*
> *1.*
> *Relating to* " Like the weighted quantile,this feature would require that*
> *the values of a given range either be presorted or sorted as part of the*
> *computation of a mean. "**
>
> *There's no need to sort a whole range to do a trimmed mean or weighted*
> *median. For the first, it's enough to do two calls to nth_element (fast)*
> *and benefit from that it partitions out the outliers without need for full*
> *sorting. Perhaps there's even an algorithm - eg a*
> *generalization/reformulation of select/introsect that does it faster than*
> *those two calls (would be interesting to know). In any case it's better*
> *than full sorting.*
>
> *I agree that trimmed mean should ideally be done by combining mean with*
> *general purpose algorithms (like nth element) but I don't know how to do*
> *that with ranges (I wouldn't - I'm new to them).*
>
> *There's also a more efficient way to do weighted median without sorting.*
> *For that I think a specific algorithm would be very useful because it's*
not
> *possible to create it out of composing other existing and proposed*

replace sort with use_n element

can we do harmonic mean in a single pass? yes there is a way
accumulator objects can do the trick, these do not calculate
median/quantize which need intermediate calculation with unbounded memory
Johan concerned that this may not be best way to do quantized, they dont
say anything about numerical stability, this may be unusable in some use
cases
1. accumulate have execution policy, so can execute in multiple threads,
simd has no order guarantee, (this means result may be unreproducible)
C++ does not guarantee order of calculation
Use Box-Muller transform instead of Gaussian transform
to get control of order of intermediate calculations, then need to roll
your own, or impl can build a more stable version so its QOI

Order of complexity mentioned in the paper (should be mostly linear)

>
> 2.
> Another general point (obvious but worth considering in the design or
> discussing in the proposal):* numerical stability can be an issue* with
> many of the defining equations being far from best practice when
> implementing. Eg,
>
https://dbs.ifi.uni-heidelberg.de/files/Team/eschubert/publications/SSDBM18-covariance-authorcopy.pdf
>
> It would be good to say something regarding that in the proposal. Perhaps
> closely related to point 3 below.
>
> Again, I trust you are more knowledgeable than me, but even summation is
> sensitive to the order of the values. Mean of (doubles) mean(eps, 30,
-30)
> vs mean(30, -30, eps) and similar with higher modes and equations.
>

> *So, it would be good if the specification of the methods allow the reader*
> *to understand how to deal with that. For example, if the sum is*
> **specified *to*
> *be added from the beginning to end to a value of the same type without*
> *extra variables to deal with numerics, the user could arrange the values*
in
> *an order (for example sort on absolute value) to reduce the effects.*
>
same as above
> *3.*
> *When the statistical distributions got into the standard they were not*
that
> *tightly specified, so we got a few unnecessary things like this (they just*
> *differ in the output *order* of box-muller)*
> *https://stackoverflow.com/questions/38532927/why-gcc-and-*
> **msvc-stdnormal-distribution-are-different**
>
> *It would be good to specify a bit more where it does not hinder the*
> *implementation. The above seems a bit unnecessary but it's not an easy*
> *trade-of vs faster or more accurate vs predictable and "same".*
>
> *4.*
> *Standard Deviation and variance*
>
> *I find it's important to make it possible to specify ddof as is possible*
> *with numpy <*
https://numpy.org/doc/stable/reference/generated/numpy.std.html*>.*
>
>
> *Doing that, it's possible to get*
> *(uncorrected) sample standard deviation (ddof = 0.0) your equation 15*
> *(corrected) sample standard deviation (ddof = 1.0) your equation 16*
> *(approximated [wikipedia]*
> *< *
https://en.wikipedia.org/wiki/Standard_deviation#Unbiased_sample_standard_deviati
on
> *)*
> *Unbiased sample standard deviation (ddof = 1.5)*
seems useful by generalizing Data_T parameter
using the last equation in the above wikipedia article
possibly as a compile time option

also variance and std dev (like numpy)

>
> *Naturally, the same with variance (again, as in numpy). sqrt is not free*
> *even these days.*
>
> *cheers, Johan Lundberg* https://www.linkedin.com/in/johanml/
>
> *PS. I'm not sure how to do nth_element piped into mean using ranges. But*
> *just to clarify what I meant in comment 1. without ranges:*
> *#include <vector>*
> *#include <algorithm>*
> *#include <iostream>*
> *int main(){*
> *std::vector<int> v{11,10,7,6,3,1,5,2,4,8,9};*
> *auto L=v.begin()+2;*
> *auto H=v.end()-2;*
> *std::nth_element(v.begin(), L, v.end());*
> *std::nth_element(L, H, v.end());*
> *for(; L!=H ; ++L){*
> *std::cout << *L << " "; // 5 8 4 6 7 3 9*
> *}*
> *}*
>
> *5.4 Trimmed Mean*
> *The issue of a trimmed mean is raised in [41]. A (p%)trimmed mean[42] is*
> *one in which each of thep/2% highestandlowestvalues (of a sorted range)* are
> *excluded from the computation of that mean. Like the weighted* quantile,this
> *feature would require that the values of a given range either be* presorted or
> *sorted as part of the computation of a mean. As an author, Phillip* Ratzloff
> *feels (a sentiment that was echoed by the author of [41]) that one might*
> *handle this (and other similar) matter via ranges, specifically by using a*
> *statement of the form auto result = values | std::ranges::sort | trim(p) |*
> *std::stats::mean*

 Jeff Garland comments:
1. rolling stats and sliding window over the data and do that with

accumulator based version instead of recalculating it over and over again
this is not in scope for this proposal, but for future
2. stats error
stats error derive from std exception, dont throw std exception directly
should define class stats_error and show a constructor, what string
3. number type limitations
limited to C++ data type but need custom types, arithmetic types lock out
other types
flexibility vs safety
more types enable more flexibility
general types need to say what kind of operations does it support, what the
implementation look like
dont forget sq root, or kurtosis
worth looking at it
also consider harmonic mean and trigonometric functions,
associativity commutativity, exec parameter does not guarantee order, not
even double and float
JM recommend dont try in first attempt unless there is market pressure
arithmetic allows strong typedef/typeoff to work
serving the user and the implementers (who need to know what ops allowed to
call on user supplied type)
can we use arithmetic now, and later relax it?
its just the constraint that is being relaxed, so that more stuff may work
in future
make a concept for these things,
but keep the arithmetic constraint

review of Combinatorics D paper
use natural logarithm by Moses/MIT
double type will max out after 127! so its not enough, we use templates for
wide integer type

plot value vs accuracy, linear declining
how many digits in the result?

do we need a factorial class which are often divided?
Matlab allows factorials of 20million but is it a limitation of the
algorithm?
is there a concept of T? like std library functions that have no concept

Please add JM and SG19 as co-author or acknowledgment to both papers

back to stats:
quantile sorted, unsorted was the most difficult, while mean , median was easy
but unsorted could use n-element
can use on any reference
85% quantile symmetric way
how to round q with the number element? add extra param like numpy to say whether you want both
with unsorted that might add new unknown algorithm

> *Stats review Richard Dosselman et al*
> *>*
> *P1708R3: Math proposal for Machine Learning: 3rd review*
> *>*
> *PXXXX: combinatorics: 1st Review*
> *>*
> > *std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2*
> > *above is the stats paper that was reviewed in Prague*
> > *http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19*
> > *
> > *Review Jolanta Polish feedback.*
> > *http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html*
> *>*
> *2.2.3 any other proposal for reviews?*
> *>*
> *2.3 Other Papers and proposals*
> *>*
> *P1416R1: SG19 - Linear Algebra for Data Science and Machine Learning*
> *>*
> *https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk7xKikK4Yp8/edit#heading=h.tj9hitg7dbtr*
> *>*

> *P1415: Machine Learning Layered list*
>
> *https://docs.google.com/document/d/1elNFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr*
>
> *2.2.2 SG14 Linear Algebra progress:*
> *Different layers of proposal*
>
> *https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXIK_d0ljdUAtSQ/edit*
>
> *2.5 Future F2F meetings:*
>
> *2.6 future C++ Standard meetings:*
> *https://isocpp.org/std/meetings-and-participation/upcoming-meetings*
>
> *None*
>
> *3. Any other business*
>
> *New reflector*
>
> *http://lists.isocpp.org/mailman/listinfo.cgi/sg19*
>
> *Old Reflector*
> *https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19*
> *<https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/sg14>*
>
> *Code and proposal Staging area*
>
> *4. Review*
>
> *4.1 Review and approve resolutions and issues [e.g., changes to SG's*
> *working draft]*
>
> *4.2 Review action items (5 min)*
>
> *5. Closing process*
>
> *5.1 Establish next agenda*
>

> *TBD*
>
> *5.2 Future meeting*
>
> *April 8 2021 02:00 PM Eastern Time (1800 UTC): Stats*

Minutes for 2021/05/12 SG19 Conference Call

Attendees:
Michael Wong
Richard Dosselmann
Will Wray
Oleksandr Koval
Scott Moe
Scott McMillan
Ren=C3=A9 Ferdinand Rivera Morell
Larry Lewis
Phil Ratzloff
Joe Sachs
Luke D'Alessandro
Andrew Lumsdaine
Ritwik Dubey
Johan Lundberg
Guy Davidson
jonan lundberg (desktop)
Marco Foco
Notes:

14:01:47 Hi everybody. Can you guys hear me.
14:01:53 Alright, sounds good.
14:01:57 Let me see if I can get him, he could hear something here well.
14:02:08 Okay,
14:02:15 looks like this version of zoom now is doing automatic transcription for us. Yes, I'm experimenting with this, I enabled it so that I can see how it works and see if this is something we want in a cell meetings.
14:02:29 I tried it yesterday with sg 14.
14:02:41 And it worked really, I mean the word translated was really good and I actually went back and started cleaning up words, although I don't believe it understand anything other than English.
14:02:45 Bonjour.
14:02:46 It shouldn't.
14:02:46 It shouldn't go me of mine here as to me allied Nicole my Yeah.
14:02:53 No, it doesn't understand Chinese German or French any other language.

14:02:58 Language.

14:03:05 We all have to store smells

14:03:05 bad.

14:03:09 bad. I can try Spanish if you want. Yeah, Go for it.

14:03:14 All at all those sucky

14:03:20 piano.

14:03:22 I cannot stand a student doesn't seem to get it so

14:03:29 so much fun we have, we can have with an automatic translator.

14:03:37 Oh, my God.

14:03:40 Hey guys, I'm, I'm going to be in a little bit of trouble chairing this call. And I wonder if I can prevail on somebody to chair it for me. And maybe guy, because he's had a little bit of experience yes actually a lot of experience yesterday with 27 people.

14:03:57 Would you be able to do another cheering for me. I'm afraid I'm having my dinner right now.

14:04:06 I ordinarily I would actually be charged by.

14:04:09 I'm absolutely styling.

14:04:14 I could take over in half an hour. No, no it's okay I don't look at other people maybe Renee on, or Larry, I unfortunately have to work at the same time.

14:04:24 Okay, let's see I think I'll Richard he's been doing this for a while.

14:04:29 I would love to but sadly I actually have another meeting starting in an hour and a half.

14:04:34 Fortunate reality the online world. Understood.

14:04:37 Anyone else Larry, would you be interested.

14:04:47 I'd be, can you hear me. Yeah. Okay, I'd be happy to do it but I have no idea what to do.

14:04:47 This that there is an agenda that I put out and all you have to do is share screen on it and then drive it along the way.

14:04:55 Well that may be harder than you think.

14:04:57 This. There's always time to learn so I'm going to tell you what I'll do this first of all make your calls and see if that'll enable a whole bunch of new buttons for you.

14:05:09 This is Larry is going to be presenting some, some stuff to you right, of course, yes, he will have to, you know, definitely have to have to do that so actually I'm going to enable share screen for everybody so that they can do that.

14:05:21 so I think today's main agenda is we got some really valuable feedback from a couple of people Alexander's on, I think. And who else was giving feedback to, to, to your document on stats.

14:05:39 So stats has not been reported yet so oh I should actually go through the normal process while Larry is setting up.

14:05:58 Yeah, I'm gonna go find my, please go find my agenda document that I sent out the agenda email I sent out yesterday was Tuesday. It will be in your email somewhere.

14:06:00 Yeah I forwarded that on to you learn.

14:06:03 Okay, hold on.

14:06:04 Okay, so.

14:06:08 Okay. It starts out with roll call participants adopted Jen Yeah, yeah. So you shared screen on that try sharing screen on that document on your computer.

14:06:16 Wait, wait a second, I'm looking at that.

14:06:26 Okay, so what I'll do is I'll do the roll call. Now since we are already keeping track of roll call on the zoom report. I don't actually have to write anybody's names down but I will call them out I see Larry lewis i see guy Davison I see Joe Sachs, I

14:06:35 see Johan Lindbergh I see Alexander Coble I see Phil rats LA I see Renee Rivera, I see Richard awesome and I see Scott McMillan, I see.

14:06:44 Scott Moore, I see will Ray and Michael long, and this is the May 13 sG 19 call.

14:06:55 So there's also a list of ISO code of conduct guidelines at the top, that people are instructed to read and to acquaint themselves, basically all it says is be nice, be decent to each other.

14:07:10 and then we'll all get along.

14:07:16 I think for today's call. We are mostly going to be going through the feedback for Richard Dawson in some stats paper which, which was published on a cell and but it has not been scheduled yet for LEWG, but they know about it, they have a lot of things

14:07:32 on their plate so they're just waiting to schedule it. But in the meantime, there's actual feedback from people read the paper, which is great, is we don't, we can actually make some adjustments on the fly already, and that's fine.

14:07:46 Sorry wasn't to the reinforcement learning.

14:07:48 And there's also reinforcement learning Yep, that's right, I'm going to get to that. And so, but I don't know if Larry and these guys have prepared anything yet.

14:07:59 But something, actually, which I have, which is, is a work in progress, but which I definitely want to discuss a little bit. So, I'll do that at the end after we get through all of those.

14:08:11 So I think it's more important to get through the feedbacks right away. Be from Johan Lindbergh and from Alexandra Kobo.

14:08:20 And then we can end up, go I suspect this meeting might be short, it'll probably be an hour anyway. So, and then after that we will do the reinforcement learning.
14:08:28 If the schedule is good with you guys, if that's good with you, Larry.
14:08:32 Yeah.
14:08:35 So, I have up the wonderful application known as Notepad.
14:08:41 Got it.
14:08:43 She wanted me to give you a shot of taking the screen here.
14:08:47 Well, you probably don't need to now because I think, Richard will probably bring up was paper. And then he can he can listen to the feedbacks, and then you just have to take notes.
14:08:59 Go, Richard. Are you ready to, to bring up the paper.
14:09:05 I can, can I just have a couple of months grab some blank paper notes, taking the lead myself I'll come back in about five minutes. Yeah, go ahead and explain the corner the procedure to everything here then.
14:09:10 Okay, so I will do that now.
14:09:18 So guys, um, we got feedback from Johan, and from Alexander, Alexander I don't think you've been on this call before Would you like to introduce yourself.
14:09:31 Yeah, I'm just developer who tried to implement those steps proposal so I am actually I'm not familiar with the whole process of like how committed works.
14:09:43 Okay.
14:09:44 Thank you very much uu, the bread and butter of how committee works is, is based depending on people like you who can do these kinds of things.
14:09:54 And Johan also gave feedback as well too. So I'll explain a little bit.
14:10:00 This is actually a little bit of an unusual group sg 19 machine learning is a outreach group, because we don't have the expertise in the committee in meaning, there is only a very few machine learning, experts are that kind in the committee.
14:10:15 We are permitted to outreach to other people who are available, much more specialist in that domain. So as a result, whereas normally you would need to do jump through all kinds of hoops to get into the committee be a member, your national body member.
14:10:30 This is all dispense with and.
14:10:33 And now since we have this pandemic thing, we can definitely do a lot of things live. I mean sorry not live but on zoom call, which means that you don't have to pay money now even better because in the past you

might have to have to pay money for a trip

14:10:44 to got to show up and all that stuff, or the very least, talk to somebody who goes there, and then have them proxy you while you can't attend because you have a job and on and you know I'm totally understandable.

14:10:57 So, having said that, this is actually the perfect time to do something like this where we can just get the feedback, look at it and then see what we can do to adjust that.

14:11:06 Having said that, though, you could definitely join your own national committee.

14:11:11 I'll quickly explain how that works because actually, I just answered a question from from, Kate.

14:11:20 On one of guys books I think about how committees work so let me just explain the committee's that ISO committees work like the United Nations, they are, they have their national delegations except get sent to them.

14:11:34 So each country has its own delegation.

14:11:38 And there's a head of delegation. So Canada is called Standards Council of Canada, the UK is called British Standards Institute, the Asana.

14:11:48 The American is called insights, or used to be called and see if there's some standards is still quite antsy.

14:11:59 But each country has its own, but not every country belongs to the ISO standard has not every country has a standard development organization. Most do.

14:12:06 And even if they haven't a standard development organization or an organization that sto might not have joined the c++ part. They might have joined what other things like elevators and and railroads, but they didn't join the c++ part.

14:12:18 So, depending on how deep you want to dive into this rabbit hole, you could go through and try to figure out, you know, which country it is that you belong to, and can do do they have an SEO, and they have an SEO does that as do.

14:12:35 That is still join the group that's relevant for c++.

14:12:40 You can do that, but I can also just tell you right now by just going to displace.

14:12:44 Okay, I'll quickly share my screen here.

14:12:49 Okay. If you go to I did you just search for a sales.

14:12:54 sc 22, which is our parent body, you'll find all the parts that belongs to the all the working groups that belongs to this, and under this thing that says participating members if you click on that you'll find all the countries that have participation

14:13:08 status and the name of the acronym of countries SPL.

14:13:13 And you'll see that these guys are participating which means they vote. And these guys are observing, which means they don't get to vote but they can show up.

14:13:22 If your country is not on any of these. That means you don't either you don't have an SEO, or your SEO never bothered to join c++, which means either way it means a huge amount of work for you to try to get involved, unless you can talk to somebody who

14:13:37 was a friendly country or something like someone like us and just go directly in. So we're trying to make those things a lot easier. Okay.

14:13:47 guy Davidson Davison Go ahead.

14:13:50 Guy Davison Go ahead. Yeah, so it might be worth mentioning that whilst it might seem like a huge amount of work.

14:13:58 It can happen quite smoothly and we only have to look at how to discover who turned up at CVP con in 2000 and right now I think it was an interesting paper on CTR and then two years later she formed and is now chairing the Czech national body.

14:14:14 Yeah, but some countries, it's not a huge amount of work for some countries, it can be a huge amount of work. and lately it's getting a lot easier. So, all Xander depending on which country you're from.

14:14:26 If you see any of these countries on your list, then you could actually join. Now having said that they are also more conditions, being part of a delegation.

14:14:35 As far as I can tell, is not doesn't necessarily mean your country of birth. I've been particularly permissive in terms of my definition I for Canada, it's basically if you.

14:14:59 We have you live here.

14:14:50 You were born, or you were born here, or your home or your company is here.

14:15:01 So those are three ways you could actually belong to Canada.

14:15:05 So I don't know which, which country are you in Alexandra.

14:15:09 Yeah, I'm in the Ukraine, it's actually in the list, and I wanted to clarify that these countries from the left side, they actually participants in the c++ so Commedia.

14:15:21 OK, so the participating countries are participating at all level of sc 22.

14:15:30 So, we don't talk about the working group levels so you can join any of these working groups within se 22 So yes, the answer is as c++ says yes, but you can also do C, or cobalt.

14:15:43 Okay.

14:15:47 And the difference between the countries on the left is that they

get to vote and the countries on the right don't get to vote yet.
14:15:53 Okay.
14:15:59 Alright.
14:15:58 All right, so let's say with that Richard, are you ready to, to, to start. I am Yes. Okay, go ahead, please.
14:16:09 So it's good probably if you can bring up your the documents so that when people talk to it, they can sort of.
14:16:18 Yeah, there we go. Thank you.
14:16:20 Hey, so we had made some revisions following I believe that was in April was last time we looked at it, Johan there from Sweden has some great feedback, they were some minor revisions, we went ahead and made those.
14:16:36 and they have now appeared in the mailing list.
14:16:40 So in the last little bit, Alexander has been so great to contact us and provide some additional feedback here that will help better this proposal.
14:17:02 And as Michael said, this thing has not been reviewed by the committee yet, which means we have the option now of making changes before they see it. So this is definitely the time to make those changes as on a separate screen here the feedback from Alexandre,
14:17:12 I believe everyone has that email.
14:17:14 I've been going through it a little bit on my own.
14:17:17 And I believe most of these are very reasonable and very logical changes to be made.
14:17:24 First one here talks about the updated constraint for the execution policy, changing the Remove CV a ref type to the decay type. No problem. Absolutely.
14:17:36 We can change that kind of thing.
14:17:39 second item on the list at any point if you need to stop and interrupt that's good.
14:17:44 The second point here the stats error should have a constructor that allows you to pass a string to it. Obviously to carry the information about the error.
14:17:57 Absolutely.
14:17:57 Not one fear would be right inside of the stats here will just improve.
14:18:01 You know just. This is our, this is your hands or Alexander's feedback. This is Alexander's exclusively today. Okay.
14:18:13 So I suggested that we have an overloaded constructor here, it takes a string, in order to pass the information of the what function here, describing what the error is, of course, we should have that.
14:18:25 Yeah, so that was overlooked can absolutely be added immediately.

14:18:33 A few of the other issues here let's take care of the easy ones first and then we'll come back because most of the challenging ones are all part of a common theme here, they're all unified.

14:18:45 First of all here.

14:18:47 The use of the naming convention underscore T.

14:18:53 Is that not c++ standard when you define a new type to post fix it with underscore te was I making a mistake with that

14:19:06 looks like guy has a response for that.

14:19:10 There's no.

14:19:11 The only naming convention that we have regarding underscores is that symbols of symbols, beginning with two or one underscore reserved.

14:19:22 There is a habit on some proposal writers to create types with underscore T.

14:19:28 It's useful but it's not mandated and it certainly wouldn't require it wouldn't be sent back.

14:19:33 Luke is responsible for evolution working group is responsible for naming and they might have words to say but over the past year or so I've noticed that we're not quite so

14:19:47 intent on making sure all new types of an underscore te.

14:19:50 It's a hangover from see it's quite useful.

14:19:57 The only place I can think about that has on this party. Now, if this short tense, or using shorthand for the traits.

14:20:12 You know, I mean like, Yeah, I agree with it.

14:20:18 I don't think it's anywhere else.

14:20:21 Currently,

14:20:27 I think more importantly also, I wouldn't worry too much about the names because they will be extensively litigated by library evolution.

14:20:37 So, you know your names of suggestions like revolution will will say yes or no to them and offer alternatives.

14:20:45 So, so don't.

14:20:47 Don't worry too much about the precise detail of names.

14:20:52 Okay. Without then at the moment I guess I'll just drop the underscore key, since it seems to be out of fashion and like you said, the library team is going to have a look at that anybody minor point fortunately.

14:21:06 Let's see if there's any other related simple questions here before we proceed to the challenging ones, another one here with interpolation underscore te no problem to change that.

14:21:35 Okay, all the modes function here or the quantum files.

14:21:28 There is the discussion and the suggestion here that rather than returning.

14:21:35 An interpolated value, such as in the case of a quantum file. What if we instead returned interpreters to the values and allow the user then to have access to them inside of the data structure.

14:21:49 I believe I originally had done this in the first version of the paper, and yen's Mauer spoke up and thought that was a poor idea, we should return values rather than each Raiders, even though for example the max element returns an iterator, I believe

14:22:04 it was the ends who suggested know we should return values, instead of iterating hours.

14:22:12 Anyone have a comment about that I was following a yen's advice believing him to be more experienced in this than myself doing me in the projected values for the raw values.

14:22:26 I think it was just the raw value, just the entire structure itself, whatever it is,

14:22:36 but it can be actually quiet enough in efficient I mean, like in the actual role Hello can be pretty big

14:22:52 that's definitely true. It looks like I had a comment about this.

14:22:58 I was only going to affirm that yes it's certainly worth listening to.

14:23:05 before you make any changes to the paper.

14:23:05 If you said something then Dublin, Ireland, and make the point, otherwise so somebody come back and tell you well no I said this.

14:23:14 And he does turn up at all the wording groups that core groups and the library wording groups, so he does carry considerable way because he is extraordinarily experienced and a very accomplished at wording.

14:23:30 Actually, so my, my take on this.

14:23:34 I started to write stuff about these but that just turned out to be too much I hate cut it out from my other comments paper, and maybe I'll write something about that I think this.

14:23:45 I think the chron tiles and mediums are actually so complicated, compared to the rest of the rest that they really

14:23:58 mean the expectations and use

14:24:02 may differ a lot, but if we want to provide something that looks like. What does in other languages like Python or statistical tools, then it has to return to some kind of value.

14:24:18 And I would, I would feel that it could be very surprising it but it's not a projected value.

14:24:25 And I think it costs a little bit against this pattern to have to return the values in some cases for this other one is optional, and all of that. So, my, my summary of the comments I haven't written yet would probably be that we should pick either either

14:24:50 be really a mass like function where you get an answer that you can use, or like the proposal I started to write for like a generalized element instead, where you actually get full access to everything and then you can do whatever you want but it's super

14:25:09 complicated to use.

14:25:12 I don't agree that we don't want to use work with iterate iterate there's us out here because you want to. We want to have a mean and median and quintiles have the same output the kind of interface.

14:25:33 Wouldn't it also mean that,

14:25:37 you know, because we need to order the containers at times. Now, to get, like, mode.

14:25:48 We would we be exposing an iterator for an internal container.

14:25:56 And so that adds additional complications.

14:25:59 I that's actually where I started with my critique and the ideas are regarding this this paper that I thought like, why isn't it done.

14:26:11 This completely other way, but I don't think that it's the way to go because and also no other.

14:26:18 No user would expect something like that either you'd have to have a very complicated.

14:26:24 You would have to return, like where to where which points, all of the points you interpolated from what interest rates did they have what weight, do they have.

14:26:37 You would have to return, like big structure with information for someone to actually be able to use it.

14:26:45 So I think it's better to just make it a black box.

14:26:50 And then you can have an algorithm, like a generic algorithm that does generic things, but they can do that is not this.

14:26:59 And they could work on whatever they thought you have, you don't have, you can.

14:27:07 Yeah.

14:27:08 Right, so we're kind of wanting to work with a scalar value and if you've got a writers then, now you're talking about dealing with ranges and multiple values and it just doesn't feel quite the right thing.

14:27:29 I help him.

14:27:32 So, as far as I understand we agreed to return value returned values and not the directors and not expose the useful, possibly useful things shuffling that happened because it could be too complicated you expose it.

14:27:51 Yeah.

14:27:54 Okay, so it seems like we still have consensus that we want to return a value. I agree, perhaps now we should probably return the

projected value, rather than the whole structure.

14:28:04 If it is like a structure that has five integers let's return the integer inside in question, rather than the whole structure that change could be made.

14:28:13 Yes.

14:28:15 I think I should write down my thought about updates about these things because I thought about what happens with you. you want to.

14:28:27 If you want to keep this by your return an optional for like the right

14:28:36 like one of the two. In case of median. I'm not, I don't think that's.

14:28:45 It would be nice if we could just avoid that, and instead just return a interpolated value, always.

14:28:52 Yeah, I think the first version we did that and somebody corrected me and thought, No, you should give the option of both.

14:29:03 Yeah, but then you have to, this is like a super special case where are you now.

14:29:13 That's the syntax around it becomes pretty annoying I think because do, and and you can only, only do it for media and not for the others because then you would need, like the remainder.

14:29:26 Wait, parts of it.

14:29:29 That's right and you always have to have an F test er there two values yeah and then you have to write your own averaging or interpolation all the time.

14:29:39 Yeah. So So all of this now on hijacking this whole discussion, sorry. But, so, so after like trying to argue in my head that we should this proposal to change to allow arbitrary iterator stuff.

14:29:52 I concluded, no, this this definitely not the right thing because you want to just use it as mean, or whatever.

14:30:00 So, So that's, I wrote.

14:30:03 It works exactly like elements, but you put in a range of me instead.

14:30:22 So, if you want to do stuff like partition and interpolate arbitrary things like records with whatever. And maybe you don't want to interpolate using floating point or anything you want.

14:30:38 and you can use generic algorithm instead.

14:30:42 OK, OK, OK, that's an alternative Yes.

14:30:46 I think we can all agree that quintiles and the median is perhaps the most difficult one here along with the mode, or maybe not but definitely those Quantel suppose a lot of problem, right from the beginning=
.

14:31:02 I have a question.
14:31:04 So,
14:31:07 The median and quintiles to me are fundamentally a little bit different. Okay, that's, that's absolutely the case yes yeah I mean, the media is mostly one or two and, you know, most of the time, you know, it goes back to you want to make the common things
14:31:24 and the easy things easy, right. Yeah, anything worse than that.
14:31:30 And the default.
14:31:32 You know, and anything more than, non common cases, you know, you want to make it an A possible do but it can be a little bit harder and almost feel like
14:31:43 different functions. But quintiles are going to return multiple values all the time.
14:31:50 So, you know that that's kind of a different case to be it's going to be a vector right i mean that's the way it's going to be or, you know.
14:32:02 So to me, those are those are kind of two different cases.
14:32:07 The way I deal with media and separately from Quantel I guess.
14:32:13 Our.
14:32:14 No. Yeah,
14:32:17 go ahead.
14:32:19 I'm just thinking about the case when the user for example has a large collection of the items, and tier one get item with the medium price for example.
14:32:30 And when we if we turn on a projected file if he can answer this question, he would need one more pass to find this item with the medium of the meat and actually.
14:32:42 Yeah, that's actually, if you want to do that. Yeah, you have to pretty much use elements.
14:32:51 I think because because you need otherwise you need to get the weights, and that depends on, depending on how we designed this on the like this scheme.
14:33:05 Like the quantum scheme, there are all of these different versions of contacts that people use, and actually could even, it's even the case I think yeah so in some cases you don't actually need to calculate all the
14:33:24 same, like partition points depending on which one the schemes you use, so it's not only a question about waiting.
14:33:32 But, and also the way you would wait, like, if you want to like actually get access to the records.
14:33:43 And that's not something you can do while interpolating really than you would think.
14:33:51 I would say use yeah their algorithm system.

14:33:55 This stuff looks like math to me.
14:34:02 Do have existing generic algorithms that can do that.
14:34:07 This is my other. This is my proposal.
14:34:16 It's called 20 375.
14:34:22 It's just I'm just writing it right now but you can you can find it.
14:34:26 It's just a generalization of elements.
14:34:30 That's all it is.
14:34:32 But that's but if you want to do things like that, then you have to do it.
14:34:47 You have to decide how do you interpolate strings, I'll do interpolate time duration so I'll do interpolate post addresses.
14:34:48 You can't really solve that.
14:34:51 I think in this setting.
14:34:57 But I think that we have like a projections for arithmetic values for doing that so we don't need to interpolate the whole strings, at least for now, as a retargeting only arithmetic types when this proposal.
14:35:12 Yeah, but then why do you need the records.
14:35:19 Well, that's just one of the use case I can't imagine like we have some records, and they all have prices, and we want to find record. For example, with the median price.
14:35:32 But that's, that's an element.
14:35:37 Actually, so that's an element that the only difference between elements.
14:35:42 And these contacts, is that you get help in calculating like how many things are in the collection, and you get how wasting them together.
14:35:56 Okay, now, Now I got to find. Okay, thank you.
14:36:03 Okay, that's actually a very nice answer suggests that if you want access to the element you should be using something like an element, or your hands new proposal this 2035 2375 instead.
14:36:15 And we should make this a math pay proposal where we just give you that value that you're looking for.
14:36:22 In which case, maybe we do want to start interpolating or averaging them together, get rid of that optional and just return one value, just like you would expect in mathematics.
14:36:35 I guess we were trying to provide both a mathematical median or quantum style, and we were allowing you to get access to these elements, when these should be separated into two different procedures.
14:36:51 The amp element ID and then the mathematical medium or Quantock keep them separate.
14:36:54 This, this took me a month of involuntary thinking to realize it's

it's really two quite different things, possibly, that we want to do.

14:37:06 Okay, well that would definitely clear it up, we would focus on the mathematics hear them and give the user what they expect like quantum or a medium, not get a Raiders or multiple values just a single medium or Quantel that they expect

14:37:24 that could definitely simplify a lot here, I think.

14:37:29 Hey, looking at some of the other points here some of the weights, being larger than one week definitely can relax that restriction and allow for that.

14:37:40 Just a brief note again and Quantel is about the sorted versus unsorted versions that was originally ends Mauer idea to separate them into the sorted and unsorted.

14:37:52 If they're unsorted you only need to make one pass across it, or you need. If it's sorted you only need to make one pass. If it's unsorted you needed this special random iterator random access range, so that you could swap things around as need be.

14:38:10 So, it was decided that those would be broken into two different things sorted and unsorted for sure.

14:38:18 The quantities for a moment and come back to them here after since there and, again, the most contentious part here

14:38:27 with other minor issues that are just technical matters to clear up and not not too difficult.

14:38:34 Again with the namings the underscore keys, we can drop that rather using a Boolean for ketosis rather than are using a parameter, rather than a Boolean sure that'll be consistent everything will be a parameter.

14:38:48 The name variants here var as a statistics person I always saw on university that variants and in textbooks was always written var anyone like the full name variants instead of var

14:39:10 mathematics.

14:39:12 So cause Did you say var.

14:39:14 Yeah, which also means, you know the difference between Mathematics and Computer Science right computer science var is variable.

14:39:24 I'm.

14:39:33 mentioned like that vs but just open civic the reference and he found the name function calls about what would he expect. I mean, the standard library is not only for mathematicians, it's for a wide range.

14:40:06 I like to longer name so suffer for so this is quite specific. Also, one could. It's. There are not that many three letter.

14:40:10 things that we

14:40:14 three that's right so far is quite a land grab.

14:40:18 And whilst far does indeed, you're correct. I also a novelist intuition, and when I see vault. That's what I see. However, the context of c++ is such that we have a lot, a lot, that identifies and var only leaves hamsters.
14:40:34 Because we're mathematicians look at the mathematical.
14:40:38 I think it was paid to assume as little as possible, of the audience.
14:40:46 Okay, okay fair enough again I was coming from the bias side of having a math background like all of us, sure we can change that to variants yeah yeah and there is a good point that yes that looks like variable as Larry says, and someone looking at the
14:40:58 standard library might think this is a variable or some sort of sure we could change that to bear needs. Yeah.
14:41:06 All of the other names are spelled out in full as well so that would be consistent then sure we can change that.
14:41:13 What do you use for standard deviation.
14:41:16 St DEVB enough for them.
14:41:21 If anybody confuse that with anything else.
14:41:26 Do we want to do a standard underscore deviation.
14:41:34 Yes. Well,
14:41:34 actually, all the, this so called special math functions are very long names, they are released.
14:41:44 I don't know info. I don't think we should be scared of long names, really, you know, lexicographical comparison. That's hardly.
14:41:52 You know that doesn't roll off the time, but don't say it means there's also the scope if you as a programmer, absolutely sick of topic and parents, you can always alias these names yourself.
14:42:05 I think the smart thing to do, sorry that's an awful thing to say no I don't mean this.
14:42:11 I think the kind thing to do, to all the consumers of C plus classes to be as an ambiguous as possible with our names.
14:42:29 Looking at the special, special mouth ones right now Riemann underscore zeta SPH underscore vessel function. Yeah, we could go ahead and say, standard underscore variants, or standard underscore TV and give the full name there.
14:42:39 Again, we're coming from the perspective of mathematicians it's clear to us but others, they might not. So, yeah, we can definitely do we could make sure.
14:42:51 A couple of other minor points here about rolling exceptions and things like that that's maybe a little bit more detail will come back to that.

14:43:00 Let's look at this big overarching idea of this execution policy something that I realized I might have overlooked.

14:43:08 If we are going to be working with an input range.

14:43:14 We're assuming you have maybe just one scan over the data.

14:43:23 If we start talking about ideas and parallel processing and things like that when you only have potentially one look at the data, there might be issues and concerns with that and that's what Alexander found when he was implementing, maybe he could speak

14:43:36 a little bit more to what he found.

14:43:40 Well, I haven't implemented thorough version of the app.

14:43:43 But at least as far as I know, like all existing algorithms.

14:43:50 With execution policy they all use for what it razors, because how you can paralyze things. When you can know the size. When, when you can read multiple writers like multiple times.

14:44:07 I mean, how can you split this range, with only inputted razors. That's true. Yeah, we haven't seen the full data we don't know the size we don't know how to carve it out.

14:44:17 You'd have to scan it once and then you can't scan it again, you'd have to save it in storage you'd have like you said an old van linear storage problem.

14:44:27 So what I was thinking about earlier today was what if we had a version here where we had two versions and input iterator if you just going to scan over something once, and there's no parallel version.

14:44:40 And then we had forward random moderators and those have overloaded execution policy versions.

14:44:49 So if you're only making one scan, you don't need execution policy, we won't allow that. But we would have overloaded versions that take input or far sorry forward it or writers and those ones could have execution policy control.

14:45:09 Just to clarify, you do.

14:45:14 Do you need random access for competition policies that would, I would have assumed that I believe only for the quintiles would be unsorted where we have to move things around the other ones I think we can get away with a forward iterator like for the

14:45:28 mean.

14:45:35 Okay, so let's talk to you need to know what you need to know. Okay, so you can.

14:45:42 Okay.

14:45:45 And, can anyone please clarify for me How can you iterate salted quanta with only input iterate or when you don't know the size of the range ahead.

14:46:05 I mean that you need to know the Quantel you need first know besides convenient to go back and pick up the right element.

14:46:14 And with the impotent writers, you can do that, and leave until you copy the whole range. This is obviously unreasonable.

14:46:24 Yes, yes, yes, that's a mistake that one really should be a forward iterator so that you can have at least one scan over at first.

14:46:34 To determine the length and then come back to the center or wherever you're looking for. Yes.

14:46:42 Looks like guy has something he can add for this.

14:46:46 Um, I wonder if the I'm embarrassing the unfamiliar with the execution policies, I appreciate how they were but are they guarantees or are they can.

14:46:58 I think we maybe could ask Michael that he would probably be the best answer that.

14:47:07 I originally did not have paralyzed version, I believe it was Michael suggested I did that it seems this year or not.

14:47:17 That's right, Mike was involved with another meeting. Okay, I know that. As far as I understand the pilot, the execution policy stuff, I mean you could you can never say that this implementation is not standard compliance because it's not the panel and

14:47:29 licensing, or anything like that you can't really prove that within the language.

14:47:34 It's not done observable, that's not observable right so, so, so quantity of connotations. As we said many times last meeting. But

14:47:47 I, I am a little bit concerned with.

14:47:53 even having a method that has to scan.

14:48:20 I mean, okay quintile is order, and, but I i would i would assume most users would not expect it to scan twice but. So, yeah. The reason why I was asking if it's a hint or a guarantee is, is this you know if these, these raise capital category is known

14:48:27 at compile time.

14:48:29 And if the parallel. If the execution policy isn't is incompatible with the Israeli category, then you know that that that could solve the problem of being pastor forward iterator.

14:48:43 When a random access iterator is needed to satisfy the execution policy.

14:49:02 Looks like Luke has something you can definitely go ahead and

14:49:10 got a wild animal there that's, that's the reality the online, we got we got a chance to wait for you, not a child but a dog, those, those are just constraints as to what is correct, right so your algorithm is correct if

14:49:26 you can, you know, the implementation is is able to do things based on

14:49:36 requirements. So it's sequential

14:49:48 also was trying to implement it I found that my implementation would completely rely on the existing existing algorithms, because all we basically need is the foremost algorithms, is just the way to combine into separately, calculated.

14:50:12 Statistics information and the mathematical formulas for this, and it should not be pretty hard. So, the final implementation at this mine condition would mostly rely on the existing algorithms.

14:50:37 I also want to bring one interesting moment, found that in this proposal execution policy diversions also are marked context for. So is this a typo or is this intentional because in theory we can use like stood his content evaluated and to completely

14:50:59 ignore any execution policy and execute and calculate everything at compile time, but no existing standard algorithm does that.

14:51:17 Okay, yes I, but I believe that is a table that is a mistake there=
.

14:51:20 I'm sorry. Could you clarify that mistake again.

14:51:25 That constant expression on the execution policy outside of the standard.

14:51:32 Okay, got it.

14:51:37 Question. Again, or something like

14:51:42 Or something like in here in execution policy.

14:51:48 Forward range,

14:51:52 potentially could be a problem since you only have one scan the data.

14:51:58 Like this. Did we want to have just a non execution policy.

14:52:04 Input iterator, and then a separate version with a forward iterator and execution policy.

14:52:29 Yes, I think, that's fine. Let's like most algorithms are implemented.

14:52:23 Okay, okay, that's what I was thinking. Earlier today when you pointed out the mistake area.

14:52:39 If, however, you have like a vector or a list and you want to do parallelization, then we would have another forward iterator for that case.

14:52:50 Okay, okay, divide those into separate cases then.

14:52:56 Okay, so that actually gives us quite a nice solution to that problem. That's good.

14:53:03 Returning again to the major discussions here, the quantum miles, and the, the medium or indirectly, right from the outset, this was the most

difficult problem.

14:53:15 Yes.

14:53:17 It looks like what we've decided here today is we want to move away from an element and allowing the user to access elements and parts of the data structure.

14:53:30 We want this to be more mathematical in nature, like Mumbai, or Python statistics.

14:53:36 So it looks like we're more interested in returning the value, rather than giving deep access to the elements kind of thing.

14:53:44 So we kind of want to merge it into one value if we can.

14:53:50 And we don't want to provide iterator access.

14:53:54 Since that would be non mathematical and you could do that with an element instead.

14:54:01 And we're still going to continue to have this idea I believe of assorted median and assorted quantum and unsorted distorted one would definitely have to have some notion of like a forward iterator so you can I get the size of it, maybe go over it again,

14:54:18 even though that's kind of an ugly thing.

14:54:21 The unsorted one definitely, I believe, is going to meet the random access iterator.

14:54:34 We definitely have to move elements around. So this one I believe needs to be the random access.

14:54:40 And we want to keep the sorted and unsorted separately. The sorted is a little bit less restrictive because it's just the middle point.

14:54:49 The unsorted has to be more restrictive. Since we're going to be using an element to swap things around.

14:54:57 I'm sorry. Within sort of version you're actually sorting the, the original container.

14:55:09 Yes.

14:55:12 Nobody ever seemed to have an objection to that I think there was a problem though some suggested that duplicating and copying was a problem so I think we agree that we would allow it to be switched around.

14:55:23 Anyone have any change of heart about that now.

14:55:29 Sorry doing mean like complete, or just like an element, sort of elements just a partial change, but it would change would change it would not leave the original unmodified it would possibly change it.

14:55:45 Okay, so let me be clear here you talked about starting the original underlying data in response to a function called sorted quintiles no sorry the unsorted the unsorted one here would be moving things around in the original data structure,

14:56:05 believe we saw another function that did something similar maybe

and that's where we got the precedents for that.
14:56:13 If I can find that while Luke speaks here, and guy answers.
14:56:19 I don't know, suggest and this may not be appropriate, but a lot of times if you can allow people to move you a container or copy you a container. If they want, you know to preserve their initial container, it would be a cost, right, not move but past
14:56:35 past your reference or move you move you a container so if they if they really don't want you to modify their data. They're going to accept the cost of a copy.
14:56:46 Otherwise, they'll just give you a reference and sort of up to them how they would like to use it now that can be surprising to some users=
.
14:56:54 So might not be appropriate here but it's an option.
14:57:01 Okay, with that perhaps the Boolean type decision choose if you want to copy it or just go ahead and mix it up it. Well, I guess what I'm saying is you're taking the range here by forwarding reference for the unsorted one note that's a random access or
14:57:19 no I'm just saying sorry I'm saying that are parameter is a forwarding reference. Yep.
14:57:24 So, if I.
14:57:29 It would be nice if there was a way for me as a color to pass you a copy.
14:57:36 If I need to and that may be what's going on here like I find know that this may modify my container.
14:57:41 I will need to make a copy of it.
14:57:44 Otherwise I'm happy to pass your reference.
14:57:46 And that may just be on a user's, it may be on the user to do that=
.
14:57:52 Okay, giving them a little bit of choice there.
14:57:54 Well, they already have the choice. They can make a copy, and give it to you.
14:57:59 Okay, okay. So I guess what I'm saying is I think it's appropriate to, if it's, you know, if it to give this efficient version to someone who doesn't care if you're if you're manipulating their container with the understanding that if they do care they
14:58:13 can make a copy.
14:58:14 Okay, just make that clear in the documentation then
14:58:19 just to verify something, my brain here I maybe I need more coffee this afternoon.
14:58:25 undistorted quintile refers to the fact that your underlying data is unsorted correct. That's right.

14:58:34 Sure don't like that name.
14:58:39 Watch I'll sell me what you're returning is inserted.
14:58:42 I would much prefer something like Quantel underscore unsorted, but that's just me.
14:58:48 Okay Quantel over unsorted data, something okay I understand what you're seeing yeah it's it's also the mutating contest.
14:58:59 Yes, yes.
14:59:01 I,
14:59:10 I was actually going to address this point that mutating and we do have things like stable partition and partition stable source and source, is there an appropriate prefix.
14:59:21 You know, unsorted content, I'm sure we can do, we can have a much longer than that if we like, really
14:59:28 good for the moon.
14:59:29 We need to use Greek letters like the mathematicians
14:59:36 SG 16th feedback. But yes, Yes we do.
14:59:41 I'm actually looking at that stage on on stage so great now in there. Yeah, that would be a good kind of naming something about Yeah.
14:59:54 Actually Python has pi has both mutating and non mutating there it's absolutely not clear, which one is which one lives in empty array in non pie and the other one lives just in line, non pie.
15:00:09 So because empty array of quantum venues, get mutated in place. Yeah, so I feel like mutated better, and when I see stable.
15:00:23 I think of an understand unstable I think of an unstable algorithm which doesn't.
15:00:28 It just means something completely different to me.
15:00:30 It has nothing to do with mutation mutation is mutation the stability is the ability it's algorithmic.
15:00:38 You can always use an exclamation point right as a second language does that Julia or somebody.
15:00:52 Seriously, I prefer mute. Me either. Some reference to mutation
15:00:59 indicate that yes we are changing the elements that are around in your data structure.
15:01:08 Okay. I'm sure we can agree again once more that quantum miles in the media year, the load the quantity was a really proven to be the most challenging, they're the most.
15:01:17 Unlike the other ones.
15:01:23 There's actually there's one question about them that I that remains, somehow, for me, then that is this
15:01:34 open close baseball thing with.
15:01:38 Does this allow people to do what what they expect to do.

15:01:44 And for median.

15:01:47 Median medium is interesting because it's just, it's not really quantity 0.5, it's integer division.

15:01:55 And there's, there's also percentiles and whatever up titles, there's all these integers integer defined.

15:02:23 Quantum tiles. And that is the thing, and if you, if you look at Wikipedia on percentile you'll find many ways, even for percentiles that are not the same as all these special.

15:02:38 floating point Q from zero to one. So, so I would, I would like to understand. First of all, I found just I think yesterday that in the site pie. There is stats m stats m compiles, which actually has parameters enough to be able to specify with to floats

15:03:05 all them, and nine times of standard kinds of floating point compliance describe that Wikipedia and this comments paper from Poland, I think, as far as I understand, you can't get those currently with compile with a compile some Indian functions in the

15:03:30 paper.

15:03:33 Okay, you're referring to your lamp those paper there yes and we spoke with her.

15:03:39 And I think we decided on this reduce set of functions.

15:03:49 I think what originally I was doing I'm looking at Python simple statistics library now I think I'm meeting the base requirements of just what Python does here not the non pie or any other library I think this was just the basic Python library meeting

15:04:08 its standard

15:04:11 01 well accepted mathematical name for the quantum file that you are calculating

15:04:22 look it up here on the Python library just says quantum divide the normal distribution and and continuous intervals with equal probability.

15:04:31 It seems Quantel seems to be that standard definition and everything else is that an extended term or specialized term for anything else.

15:04:47 Well, I'm not sure I agree with that. Right.

15:04:44 If you look at paper or Wikipedia on compile.

15:04:52 I think the, the way, non paid assets.

15:05:05 There's many many choices on, like if you want to divide by n or n minus one, or if you want to interpolate like from. I mean, there's so many kind of weird arbitrary choices you can do when you interpolate your data like do you want to consider each

15:05:19 kind of been as starting from the left. And the last been update, starting from the rights and then you kind of interpolate from the edges or

dissenters or there's um there's the one that the Python does by default is actually quite.

15:05:39 I like it, but it's actually quite different from all of the others.

15:05:45 And it's not the default in most other packages like side pie and maple and hard, and they most of them actually support all of these different ways to do things.

15:05:57 And then we have with an accepted, then people would know right.

15:06:03 I don't think it really has one it's, it's kind of it's like, yeah, Wikipedia.

15:06:11 Number seven or something like that.

15:06:14 Wanting to become a columnist

15:06:19 economists are very good at corralling terms together for each other, maybe they have a, they actually use this stuff lots maybe they have a better glossary.

15:06:31 That's true.

15:06:32 But the interesting thing I think was was his this.

15:06:36 I mean, if you think about being able to generalize this and actually make people, allow people to do what they do to push their current route, so sorry route, are or non pie or whatever they used to c++, if we, if we had something like what site by us.

15:06:58 us. It's two floating point values, and they, depending on what you set these two.

15:07:11 I mean this is, then you get all of these variants variations, described Wikipedia.

15:07:14 And this is still only for floating point.

15:07:19 Kind of interpolation. If you also go to.

15:07:23 If you look at percentiles which is very.

15:07:29 Those kinds of integer based fractions.

15:07:32 That's actually in a way, quite different, because then you can start to interpolate things that don't necessarily require flopping points multiplication.

15:07:44 So you could you could do model.

15:07:50 Some types of maybe not doesn't support dropping points.

15:07:54 You can actually interpolate in in a much broader set of corrected things then flows.

15:08:07 But maybe that's, that's something else but but I find it a little bit

15:08:16 annoying that

15:08:19 we can't. Usually we have too much almost flexibility in c++ library, libraries, but

15:08:33 I like to have this, this possibility to specify this all family today call it in, in the rooms in their site by Johann, are you referring to percentiles as divided equally into 101 100 groups.

15:08:53 Yes, but if you look at Wikipedia on how people do that there's also many ways.

15:09:01 And that's only 4% five and then there's also the divide by four and all kinds of include groups that. And if you actually read only 40,000%. Yeah, exactly and that's not the same as 25%.

15:09:18 I really would take Wikipedia with a pinch of salt when it comes to naming things of course much. Yeah, yeah, yeah but but still.

15:09:30 I mean, it allows us to discover that there is a concept of integer compiles that are actually not the same from, from a P.

15:09:40 That is defined by floating point.

15:09:48 What did we do for standard deviation. I mean there's population and sample.

15:09:54 That one we have a parameter there to pass.

15:09:57 I'm thinking with what Johan said if we were able to meet the base requirement or the base standard here set by side pie that might be all right. We would allow you to pass to floating point values, give you all of these variations here.

15:10:12 It leaves open the question but percentiles but at least we could match everything that sci fi is doing with the basic quantum miles is I believe that's enough flexibility.

15:10:24 I prefer. In this case you have any new

15:10:29 rather than infer from two floating point values if we've got nine types we should have an email with nine types of, same with me essentially the same did the same with any of these things will have alternative calculations standard deviation Quantel,

15:10:45 you know, percentile.

15:10:48 That's my.

15:11:05 half of 1.5.

15:11:03 So we started opening it up to making it more generic like that. I know it breaks mystical justification to that.

15:11:11 Historical justification to that.

15:11:22 Yeah, I would.

15:11:17 Yeah.

15:11:22 I find I find it a little bit. I mean,

15:11:31 one could have names for these. I mean one that could be, it could be one argument that has a default value, but there's no reason that because implementation is going to use these two values, there's no reason to not expose that to the user because they,

15:11:47 their values have a meaning, that's how you round up and down and how you divide the range of stuff.
15:11:57 You could, it could have some of the sign that lists these default things as I'm not sure they have names in them, they, they are really our type four or 569.
15:12:15 I'm not sure we want to refer to our.
15:12:21 The standard either. Yeah, they don't have names and then we have to ensure that we cover all the cases. Yeah.
15:12:28 Well if those two floating point values.
15:12:32 Let me ask.
15:12:35 Could a user specify, I don't know what those are. Did you just specify two values which would make sense, which would not make sense.
15:12:43 Yes.
15:12:47 I hate giving people rope to hang themselves.
15:12:54 Bye.
15:13:07 Here it is in the comments there the link to this thing, see what this thing officially looks like, really, all the, all the statistics, utilities, support these, at least.
15:13:30 And
15:13:30 that means ones like SAS SPS and Sr. Yeah, all of them had this.
15:13:38 That kind of sets a precedent or a bit of a standard then to follow.
15:13:51 Except Excel.
15:14:06 No, it's a little bit of a struggle for the visa I can definitely see that if we follow the standard set by all these other packages were at least in line with what everyone else is thinking, and it does provide the flexibility here although Yes, it's
15:14:19 not going to be an enumerated value it's going to break our standard.
15:14:24 But we can appeal to the fact that we're doing what everybody else is doing the de facto standard here.
15:14:42 Just as a really off the wall remark here, does anyone think that it is possible, or realistic that this proposal should perhaps so mid Kwan tiles in the median here for the first iteration and allow that to come in the second after there's been a lot
15:14:57 more time to think about it, given the trials and difficulties with this.
15:15:03 Or do we want to try to ship something that has that in there.
15:15:11 Is there a most common one you think.
15:15:16 I guess kind of what we're seeing in like sci fi here.
15:15:20 And then if we want to add on the added percentiles and more

flexible features leaked later, but just have this first incarnation.

15:15:29 Looks like guy has something he could say, yes, suddenly, smaller papers get through a lot more easily than large papers.

15:15:38 One approach, you could take, which we took with linear algebra was to first of all, prepare a presentation in a position paper that says there's a whole pile of special functions that we want to add.

15:15:52 Here's a list of them. Here's the all that we're going to attack this in. We're putting the easy ones first, so that we can think about the harder ones whilst we're going through the easy ones, we're much more likely to get time in front of lobby working

15:16:06 group with a 10 page paper than with a 40 page paper. So, unless they're unless there is any, you know, direct relationship between them that requires any of these functions to be pulled together, then there's a there's a lot of benefits and pulling the

15:16:29 partners into smaller papers.

15:16:27 Okay, along with kind of the indirect or promise here that they are coming, so they don't give the impression that permitting that major feature. Make it clear in this this first paper that those are coming.

15:16:38 But there's going to be a little bit more thought about the first. This is what we did within your algebra, but when I first came out the linear algebra paper people said oh need solvers.

15:16:48 How's the game, gave god no doubt, you know, we've never get a favor out if we did that. So we have this layered approach we we start with the matrix class and then we build on that and the same way you start with fundamentals and say Don't worry more

15:17:01 is coming. It'll be fine. reassure people.

15:17:03 Okay, so this one then would look at the mean, standard deviation and variance. And then the mode, the mode I think is easy enough to get in here that one's not too much of a problem, but the quantum files here, and we can appeal to this fact that we

15:17:18 have inconsistent definitions of even what quantity is.

15:17:24 There's this de facto standard that most of the statistical libraries follow but Wikipedia says something else so it's not advisable to push through the quantum at this point since there's just too much debate.

15:17:36 But we are looking at it, we're not forgetting about it. You're not leaving out this major feature that's coming in the future.

15:17:47 I can deal with that.

15:17:50 So it's the idea to spit it out.

15:17:54 The content and media from this paper, or is it to make it.

15:18:00 Did you say remove mediums well.

15:18:06 Medium perhaps if there's not too much debate about that at least

in for for almost for sure in this sort of case, because we know it's just immediately in the center their average them together, We decided we give you one value back.

15:18:21 It's the quantum file here and the definitions associated with them that's really posing the problem.

15:18:27 So you would at least for the time being have the ability to compute compute a median quantum while we would say as forthcoming just given the difficulties will give you a medium at least as a start.

15:18:41 One could argue that if we're not ready to decide on the design interface and

15:18:50 requirements contracts on contracts that could influence our thinking about. We can also buy things that we're talking about now like, what should be returned to be the prediction should it be the value

15:19:09 or not. Yeah, yeah.

15:19:11 So it looks like I would I would like to tie them together. Actually, I, I'm really interested in these I think they should come in as soon as possible but not sooner, and not the price of the other things I think, okay, so it seems like we're kind of

15:19:29 in agreement that we should take out this section, it's just we want to take out both Quantel and medium,

15:19:39 or just leave a partial media in here like for assorted case where it's very easy return an average value user has something to work with until the quantities comes.

15:19:51 Go ahead. Go.

15:19:54 This is more of a meta point I noticed the top we've been here for an hour and 20 minutes, and there is another paper that's we're expected to review here.

15:20:03 We should probably Larry your chair, which we do show a meeting coming up soon so I think I'm.

15:20:12 Since I'm giving the paper, it's not really a paper first of all, I mean it's it's really a PowerPoint when I'm not even close to a paper on this.

15:20:23 Was it was gonna say always, always do is make sure is make sure that if we continue with this we're continuing. As a matter of choice, rather than by default.

15:20:32 Right. So, at this point, I don't know what how do you should we take a vote, I guess the proposals are.

15:20:44 So, First of all the proposal to remove Quantel.

15:20:49 Let's just.

15:20:51 Can we take a vote. I'm not sure how to take a vote, so.

15:20:56 So, this is Michael sorry I've been monitoring I've been listening

to the call, but now I'm back.

15:21:02 I think, I don't know of a vote is needed. I think it's Richard has taken all the feedback, he will go off and and make the adjustments, according to what he picks out.

15:21:13 Is that the case, I'll definitely be in correspondence with everyone through email and mailing list I'll make Yeah, we have some input for sure. Yeah, yeah.

15:21:22 Yeah, yeah. So I don't think we need a boat here. Unless you want one, but that's fine that that goes ahead I'll send the email but it looks like at minimum, we're going to pull the quantum miles out, just give me an it's not ready.

15:21:26 No, no, be careful as you make the changes to the paper, because you're, you're making change to a published paper, paper, you should make the change to a new up the paper, a D, whatever this this one is.

15:21:48 Okay, the same number, but the new revision up one, and then we'll review that one. And here's the problem. When you get to LEWG which paper Do you review.

15:21:55 Okay.

15:21:57 If there's a lot of changes here.

15:22:00 Or there's only a sorry it's if there's only minor changes here, then you could potentially post that paper to the group, so that they can see that oh we have some, we had some late updates and we can, we would rather that you review the newly updated

15:22:16 paper. Okay, so hopefully that's probably the best best thing if you can do it like that, if there are a lot of changes then we're going to have to probably, if we can get it out in on the 15th of this month, or the next month, then, then, then publish

15:22:31 another paper, and that way they can always have the latest paper.

15:22:34 Okay, okay, okay, it's not going to be too terribly much it's going to be segmenting the other algorithms into. Okay, and the iterative, or input iterator and a forward iterative version, and then pulling out these quantum miles here is going to be the

15:22:48 big chain guys know Be careful though. Are you sure that these changes are blessed by everyone and everyone will like it always, always thinks that like this is what I'm changing going to increase the consensus of the paper.

15:23:02 Meaning, more people are going to like a more people going to love it.

15:23:06 If it's not clear that more people are going to love it. You don't have to make the change what you can do is put it as an alternative and say, oh by the way some people would like it like this.

15:23:17 Okay. You can do it as a new addition as an additional attendance section. So, you make the choice but be careful you don't want to decrease the consensus on this paper.
15:23:28 Okay, some of the issues don't like with that execution policy is going to be a problem if it's just an input and radio so that exactly if you think something is going to get even slightly controversial don't, you don't have to.
15:23:40 Don't you don't have to throw away but put it in as an attendee.
15:23:43 Okay, okay. I think some of those are kind of almost mistakes that need to be corrected. Okay, if there's a clear mistakes, go for it. Yeah.
15:23:52 Okay. So, as I understand that. I mean, this is, it's not this group that writes this paper.
15:23:59 It's, we are discussing feedback on the paper. Yeah, it's so we're not voting on how the paper should look or anything like that.
15:24:10 We still have to vote to exit the paper we did that already is. Exactly. It's already done. Oh, this is just back. Yeah, yeah so and that's the same so I we're not going to go through this today, I think, at least, we don't have time and I don't think
15:24:24 anyone has prepared for it. I have written this other feedback paper. Also, which talks about the exceptions and things like that. Right. But, but that is also I mean, that is that is input.
15:24:53 So, whatever happened to it will happen.
15:24:46 So Michael has his hand up but I will answer your question first. You have published a paper about exceptions, is that a piece of paper that's published.
15:25:02 Now, right, it's going to be P paper tomorrow or whenever the mailing is. OK, so it's going to be published this month. Right.
15:25:06 Yeah, it's the 15th I think the team. Yeah, yeah it's currently in SP. You want to review it here in this group.
15:25:18 I don't know what what the best way is that we don't have time for today I think when no no we have time.
15:25:25 Okay.
15:25:27 People about to publish is much more urgent than anything else we have on the left. Okay, so if you have a paper that's about to go public. We that that trumps everything else on the agenda.
15:25:39 So, in that case we would look at that paper immediately.
15:25:44 Okay.
15:25:45 And that way you can get feedback from this group, and then you then you can decide whether you want to go through with publishing or not.
15:25:52 Okay.
15:25:53 Michael, go ahead.

15:25:54 Yeah, it's more of a personal preference, maybe, but the the idea of having choices in the paper is good, in my opinion, but the idea of having too many choices.

15:26:07 I think it might be detrimental to the, to the paper so it, I, I suggest a weighted so for very important decisions to to have the options in the paper but not for every detail because that might result in confusion, at least when I read the paper that

15:26:24 has too many decisions to take. I find it confusing that I cannot.

15:26:32 Understood. Yeah. Yeah.

15:26:36 Yeah.

15:26:36 I can't believe it won't be too many choices but I just want to tell which had all the options he has. But, but you're right. Don't don't don't put so many choices that it becomes very confusing I find it confusing to to so many choices.

15:26:49 Okay, so, um, so let's take a sorry I'm just going to I'm going to join back in and help chair, because I'm back.

15:26:58 Thank you very much to Larry for driving us to choose.

15:27:02 It was great job, absolutely fantastic given that you really well you've been with us for over six months now so it's up to you. You've seen enough of these.

15:27:12 So I think we have a couple of things in front of us just to take stock, we have the Johan paper on exceptions. We have guy Davidson was going to give an update on linear algebra, I believe, and we also have the reinforcement learning feedback or approach

15:27:30 of these, I want to make sure we get the exception paper looked at and reviewed and then whatever time we have we can do the feedback and feedback or the, all the linear algebra look but if we don't have time, I would relegate those to the next meeting.

15:27:45 Okay, with that, would that work.

15:27:50 Okay. is that okay all right guys okay Larry Are you okay with that.

15:27:56 Sure.

15:27:57 Okay, thank you. Just high, it's just time time time time management here. So with that, I would say I would like to see if I can get them, Johan to show his paper and discuss it.

15:28:11 Yeah.

15:28:12 Let's see.

15:28:16 segments.

15:28:41 alright.

15:28:57 Right.

15:28:59 Yeah. So,

15:29:02 After last time when the call I, I thought, I have trouble making myself.
15:29:12 Understood. So, I started to write down my thoughts instead. And that went out of hand into this part too long text.
15:29:27 So, it's mostly a lot of References Footnotes disguised as a paper=.
15:29:32 But what I what I do in this paper if I, I think I can go through all the cases where, where exceptions are thrown stats are exception, and I compare it to other things that currently exists in.
15:29:50 In, c++, and other libraries, like, especially math functions and the existing math functions. And you even like mid range cases.
15:30:03 For example for in their products and transform and all of those we gracefully deal with empty ranges, and pretty much all the main
15:30:16 means mainstream tools for statistics, either house, have a special mechanism to deal with kind of missing values empty ranges, norms, or things like that.
15:30:29 And it's far too long to really go through, but I can, I can try to to outline, why I come to the conclusions they come to in that, if you looked at for example mean there's special casing on empty on some on sizes=.
15:30:55 There's
15:30:59 Sierra weights, and so on. And all of these, I think fall into categories of special types of error conditions that we already have in the standard library.
15:31:15 Like, whole errors, the main errors. Yes, that's we have for long, and even division by zero.
15:31:25 And so,
15:31:31 if you look at
15:31:35 this case of missing data that shows up in many cases like precocious and standard deviation all of these.
15:31:45 They are also, I find it much better to extend kind of input domain to love them.
15:31:53 And that's what most of the commercial tools and libraries do not the commercial tools, and various libraries do.
15:32:03 So for example, love, and stop by and all of these have some way to return, but it always get the value back.
15:32:14 And there's no exceptions or problems, usually in any of these kinds of cases,
15:32:25 Even then C sharp, those libraries use, not as a way of returning
15:32:33 problematic conditions, so.
15:32:48 Yeah.

15:32:48 Yeah.
15:32:50 And then also, if you think of
15:32:55 how these things.
15:32:59 If we are going to use these functions as cornerstones to or
building blocks for more advanced algorithms like machine learning stuff
for image processing or whatever.
15:33:14 And there's going to be like higher order functions operating on
them.
15:33:19 And usually mass is not expected to throw exceptions, so that that
is probably going to annoy a lot of people. There's also a large fraction
of c++ developers that don't have, except just like enabled, and I.
15:33:39 There's some statistics on that, but I would assume or.
15:33:46 I'm pretty sure that the people who are really doing the miracle
computations, have a much larger fractional.
15:33:57 We probably need to disable exceptions.
15:33:59 So, in addition to that, if you think about the freestanding
stuff, as in this, the freestanding limitation stuff.
15:34:12 You don't have exceptions. And if we completely avoid exceptions
and do us the other libraries to return, like, Mom, or another mechanism to
the return missing values.
15:34:26 That also allows it to be used on Google Docs or.
15:34:32 Most of these kinds of things are done on GPUs.
15:34:37 Now we're soon.
15:34:42 Yeah, that's kind of the.
15:34:49 There's one like side notes that it's interesting that many come
to the same conclusion us, as this paper does, which is reassuring
regarding how to deal with integer inputs like even like
15:35:10 ups, or.
15:35:14 Let's see what else I think like, even the pythons, much like
shows they convert into your inputs expected double what floats are kept us
blows.
15:35:27 And I find it interesting because that's the same the same as
proposed in the paper.
15:35:35 That's, I think the only way to kind of actually make it work. But
it's interesting because kind of makes it not a generic kind of thing.
15:35:59 It's really kind of a special case in there extolled the work
within yourself.
15:36:01 Yeah so
15:36:04 conclusion is basically try to return. Mom, or in exactly us
existing math functions to go through each of the functions of specify what
is the safest hetero value to return in each case, for example, if you're

if you return, like.

15:36:31 So the most common is to return nom or sample, the causes of the few values and that is really.

15:36:43 I mean, for example at my company we're always right, we have our search, have a special rules relating to how you write your research you always have to be with the possibility of Norman and the search.

15:36:56 If you, If you keep things.

15:37:00 You understand how to work with non, then this is extremely powerful.

15:37:08 Since we are only actually only working with really supporting talking points.

15:37:16 I think this really makes sense.

15:37:24 Yeah.

15:37:24 I think that's the summary of

15:37:28 the paper.

15:37:31 All right, questions from people.

15:37:39 One of the things that we've we, in fact, A lot of things in SaaS we've.

15:37:49 And I've been the one to introduce c++ at the core of where SAS has a lot of their analytics.

15:37:59 And in fact that a part of this they do turn on the exceptions, the frying pan exceptions to catch these things and it's been something that we've had, you know, we're kind of in process of dealing with, I think, in fact, one of the things that we've

15:38:14 had to deal with is the difference between what the existing C code does vs GPU code, which is just to evaluate some of these things. Some of the things as a Nan within GPU because it doesn't have the concept of floating point exceptions and Larry is

15:38:42 not pushing on this one, a fair amount.

15:38:48 pushing on this one, a fair amount. And to be able to essentially, turn off the, the floating point exceptions because within the seat code, what they'll do is they'll, they'll catch that and then try to do something with it.

15:39:02 And so I think what I'm doing is I'm

15:39:08 endorsing what you're saying that we shouldn't be. And there's also a little bit of a confusion between c++ exceptions and and hardware exceptions. And I tend to refer to the hardware exceptions as more dealing like signals or hardware signals to help

15:39:31 separate the terminology between those two.

15:39:39 And I don't know.

15:39:42 I don't have a sense of how other companies are or what other

companies might do in terms of turning on the hardware exceptions to catch these things.

15:39:57 Do what and what are their.

15:40:01 The,

15:40:07 the stats exception being used, I don't recall in what cases are those expecting to be thrown.

15:40:17 Yeah, I can share ideas share again.

15:40:22 It's really these categories, it's empty or to label data

15:40:31 or coal, other kinds of it's kind of the same thing as division by zero so I just clump that up as a poll error because eventually leads to division by zero.

15:40:45 So the sum of the weights, is equal to zero. That could either be a precondition and would deal with it as other preconditions like contract and that contract.

15:40:58 I mean, we don't show except shows up on contracts. Usually, and.

15:41:11 And then,

15:41:10 yeah, so this is the domain error, like, square root of minus three.

15:41:17 It's matching sizes of two inputs like rates and values that is also contract violation in other algorithms, I don't think it should be an exception here.

15:41:33 Okay.

15:41:41 Yeah, go ahead. Look. Now that's it.

15:41:39 I just wanted to clarify I heard two different two different things.

15:41:43 Phil was talking about hardware exceptions. I think the original comment about not having exceptions available was with respect to compiling with F no exceptions.

15:41:53 Not anything At the hardware level.

15:41:57 Am I right,

15:42:04 there are

15:42:08 there are different things.

15:42:14 queued up for example. Currently there's no GPU that supports a hardware exceptions. Then there. Then we have the people who turn off c++ exceptions but they may still have like floating point tracking stuff going on, or on Windows you can turn on c++

15:42:28 exceptions but still have the same mechanism similar kinds of exceptions that signal out of bounds reads in the stack and stuff like that. But, and then you have these, which is called.

15:42:46 I mean, this math error signaling, which is math error handling.

15:42:54 It's actually when describe that to see that the reference.

15:42:58 Bottom well described it lists all these different kinds of conditions, and you can in your implementation, decide if you want to have like signaling, and how we want to possibly catch that somehow, but yeah.

15:43:17 No, I understand all of that what I was referring to is your original statement was that there are statistics that show that people, people do not have exceptions available I thought that was in reference to people using c++ exceptions not hardware level

15:43:30 signal later yeah yeah that's turning off the exceptions from there.

15:43:34 Okay. And I don't think that's what Phil was referring to you're not trying to run exceptions on the GPU or anything so, Yeah, I think we're in agreement

15:43:47 and guide Davidson, please.

15:43:51 I'm just thinking a bit of clarification here about half no exceptions. And I've been meaning to ask this for some time now. When I'm a visual studio user I used to the marks official C compiler, visual c++ compiler.

15:44:04 And when we turn off exceptions what we're actually turning off is exception handling. So exceptions are thrown but they can't be caught is is F no exceptions, which I presume is climate GCC difference in that way does it actually forbid throwing because

15:44:18 for example, all my current exception me is turned off I can still throw it just caught by the.

15:44:27 If the translation unit you're using has a throw it will copy it will not compile.

15:44:32 Right. Okay, so if you link to something that has throws, then it will just, it will just terminate.

15:44:40 Right, so it sort of depends on which point you're in but yes it will prevent you from using throw in a translation unit where it's enabled, and it will cause unexpected results Hey them to something that had.

15:44:53 Okay, thanks for lunch.

15:45:00 Okay.

15:45:02 More questions.

15:45:09 I. Are you, are we comfortable with publishing this paper so is it just to make sure I understand. Let's say you're, you're adding exceptions to the error handling.

15:45:22 In this paper it's a friendly amendment really to the stats paper.

15:45:30 You could say that it's a hostile amendment, because I'm trying to remove exceptions from. I'm trying to completely remove the stats that are exceptional from the paper and replace it with specify the return values in all cases, I think that makes a lot

15:45:48 of sense. I think it would make it much more compostable you would be able to write log of

15:45:59 ketosis. And you don't care if it's the log or if Potosi starts with us.

15:46:12 That gives you some by Sierra or you will just get step number if no matter what.

15:46:14 So, so that I would not say it's a, it's, it's up to you to decide, but it's not family, I think, Okay.

15:46:25 Has Richard looks at this

15:46:29 person doesn't

15:46:33 or doesn't mean it's it's just off the press, so it's Yes, I haven't heard any feedback and I don't think right no one else read it's really.

15:46:43 So what's going to happen is if we publish this, they're going to basically asked for feedback from Richard, because this is really against his paper.

15:46:54 I think that

15:46:58 if you can hold off another month, and have Richard, give feedback on this.

15:47:05 That would, I mean I can't stop you from publishing I think it's fine but it's if you can hold off another month and wait until the next 15 which would be June 15, then that means that we'll have another cycle, and look at this and get riches feedback,

15:47:19 then we might be able to

15:47:24 say that this should go. This, this, because with that then we can add his feedback to the paper as well you can you can summarize this feedback and the feedback from the group and add it to the paper.

15:47:38 I got it totally wrong. I originally thought this was a met friendly amendment, but because it's a it's a hostile amendment. Well that's going to happen is they're going to come back to us and ask what did you, what did you think of this.

15:47:50 And if we say well we haven't had time to look at it, then we'll look like we haven't done our job.

15:47:56 So, I think, I mean I have, I have joined this very late so that's okay yeah that's just how it is. I mean, that's just how it is. But I would, I would really like to just to put it out there, because I think I'm kind of done with it, and I'm not sure

15:48:17 if I can practice. I mean now I have some energy and time.

15:48:30 I'm not sure what happens like what right. Okay, if you put it out there. I'm SG six will look at it, or while they don't actually look at it because they don't have any online meetings, so they will look at it, it

will come back to SG 19 which is fine
15:48:40 you can go ahead and publish it, which means that we will look at it again next month. And with Richard here and get feedback, and there's nothing there that's that's progressing to the next stage, which is also fine so I think I'm fine if you want to
15:48:54 publish it. I'm just saying that if you want to hold up that's also fine so I want to see what guy says Go ahead.
15:49:02 I just want to speak and supportive, your hands favor.
15:49:06 I agree that's Richard I think Richard is is overusing exceptions and the sort of way that has cancelled against in 0709.
15:49:18 Setting exceptions. I think that exceptions maybe being used as as yet your ancestors were pre and post conditions would be more appropriate contract violation to be more appropriate, or simply undefined behavior if the if the user supplies the wrong
15:49:31 inputs, then we should be throwing exceptions, we should be just saying that's that's on you.
15:49:37 Okay, good feedback.
15:49:41 I think in that case and.
15:49:44 Are there any other feedback so in that case I'm going to ask for vote then on this paper to go out and.
15:49:52 And if it goes out that's fine because the heading says, Come to SG 19. So that means that next month you have to come back anyway you're. Hopefully you can, yeah.
15:50:04 i.
15:50:03 Yeah. To be honest, I didn't expect to be able to have it all, I thought, okay, I'll maybe I'll get it off to the 15th and just have it out, and then we'll see what happens.
15:50:22 So, so, having the possibility to discuss it here. Let's save us some time but I still actually don't know that I have written.
15:50:24 Okay, good. All right.
15:50:25 Mike only. Yeah, go ahead.
15:50:29 Wouldn't be recycled.
15:50:34 Yes, absolutely. We can start it right away. And, and it could be it could be ongoing publishing this does not mean that this is a done deal. It's just first iteration.
15:50:47 So you've already got a number it's three p 2376 or zero, and that's fine. And since you're targeting only these two groups, it means that library evolution won't grab ahold of it yet, and that's that's that's good, that's that's what I want.
15:51:04 That's that's what I want. Exactly. That's what I want to, I just want them to grab ahold of paper that that's that's, you know, so, until we

say that okay yeah this, we can say what are we integrated with Richard style or not that's fine.

15:51:15 Okay. So, are there any other discussions and questions on this, we actually only got about nine minutes left so we definitely won't have time to get to anything else I think I would like to just, I mean, I assume, no one in here actually read it.

15:51:40 Really yet, because it's just came out, maybe someone did that. Bu=t

15:51:53 I think I explained myself better text them dumbing

15:51:45 them in a foreign language.

15:51:50 So, so it's easier. Yeah, so, yeah.

15:51:52 Okay, no you, you did a great job, no worries that there's some procedural things that I was just trying to enlighten so.

15:52:00 So what is this paper number it's p.

15:52:06 272376760.

15:52:13 He published

15:52:16 for sg 1966 right.

15:52:25 Okay.

15:52:32 I just want to ask a question.

15:52:36 Yeah, go ahead. Go ahead, ask your question, we are insofar as how this paper affects the stats paper we aren't actually therefore suggesting any changes in the stats paper at this point.

15:52:47 No, no.

15:52:51 But there might be changes that's going to be negotiated if we after our review next month.

15:52:56 Okay. Yeah.

15:52:58 Because that until we enact it and accept this paper we won't obviously yeah yeah hmm. If I I'm I'll chat with Richard, on occasion. So do you have any problem with me just giving him a heads up on this.

15:53:16 Just let him know that you've got this in process, no no that's.

15:53:23 It is this on mentioned on the mailing list and it's okay.

15:53:30 Okay, so this is mostly.

15:53:35 I'm not entirely sure this is the, we don't even if you're going to, I mean we don't probably don't even need to really pull but maybe to useful for you though, I think you'll find is, whether you think, I think, Okay, you know what I don't have a better

15:53:47 point I was going to say, move that this paper be published but you know what we don't need that Paul. I think you want to get a poll how many people feel that this is a good direction.

15:53:57 That's that would be nice. Yeah, yeah, yeah, definitely. I mean, because I'm.

15:54:03 I mean, just to clarify, I mean, I don't expect to write anything like updates on this paper. I mean, so it's not, it's it's the stats paper that promotes this old stuff right so so feedback is eventually going to be=
.
15:54:22 To do that, but I would of course want to know what you think that is.
15:54:27 Yes.
15:54:28 So let me see if this work.
15:54:35 Did you see the pole.
15:54:37 Does everyone see the pole.
15:54:53 I don't
15:54:45 know I don't
15:54:48 know you will pull away I have to do this at the launch the poll. How about now. Yeah.
15:55:02 It's happening.
15:55:09 People are changing choices. It's cool.
15:55:12 Wait, wait, I got it. Can I feel like I don't see where I vote. I'm sorry.
15:55:18 Oh, because no you can't vote because your hose I'm gonna make you a non host, I believe.
15:55:24 Okay.
15:55:25 How about now can you see it. Yes. Okay, as a host I cannot vote. This is the job back on this system, but I'll just add just add myself in that one of the numbering system.
15:55:36 so I got five out of 12 strongly for.
15:55:39 Oh, is everything done, I think I can publish the results right.
15:55:43 Is everybody done voting.
15:55:46 Well I see the pole but when I, the submit button is grayed out. Oh well, I was I voted I voted 100,000% strongly yes, even the strike guy.
15:56:01 Um, are you able to show us. I'm not from mom and dad and honestly Michael but I'm wondering if you're able to show us the results of the poll. I'm going to try this is all an experiment you know I love experiment=
.
15:56:14 So, this is absolutely fabulous but, you know, follow me say Oh, these are the results and support.
15:56:19 Thank you very much. All right now going ahead and pulling and let's see what happens okay guys.
15:56:28 You guys see anything.
15:56:31 Well, there it is. What happened, what do you see, I don't want to tell me what you guys see results, 46%, probably for 38% or 15% mutual, not simply begins no concern strongly against, that's what we see.

15:56:47 All right.
15:56:48 Get tested century.
15:56:54 No more racing and no one knows what.
15:57:00 Okay, so do we have to add any, any votes that wasn't cast.
15:57:03 Larry said you didn't you didn't cast your vote right.
15:57:07 Yes, my Submit button was great out I can't.
15:57:11 Oh, it probably was because you were a co host so you casting strongly for.
15:57:16 I was like in the category right above strongly for okay so yeah so that makes it seven out of 13.
15:57:24 Yeah, we don't get to see the numbers here doing, how do how do we assess consensus with numbers.
15:57:36 well we don't see how many people have stained was a massive post and he was absolutely you'd have tracked it right because six people votes. Well seven people to strongly for five weekly for, we didn't, we didn't see those numbers and you see the numbers
15:57:46 what we said you don't see them. Okay, we only see this, I don't I do see the numbers, and I'm going to download the results and see what happens. I want to see what happens like download the results.
15:57:59 We obviously can't handle the actual raw data.
15:58:05 This is thrilling thrilling thrilling.
15:58:08 So what I saw. How about this, I'll do this I'll do this, I'll do this guy's I'll do. I'm going to I'm going to share screen.
15:58:16 And this.
15:58:17 That's, yes.
15:58:19 Okay, this is what I see on my screen.
15:58:22 You guys see this.
15:58:26 No.
15:58:27 Do you see the pole. No, no.
15:58:31 I see that you have many tabs open.
15:58:39 He's trying to read the entire internet.
15:58:45 You guys and I'm not trying to impress you with the amount.
15:58:51 But I think Michael, you know, probably has special ramp just for the tabs.
15:58:58 All right. We can't see the Paul Michael I'm sorry okay so you use, somehow it's only.
15:59:08 Um, yeah, so you guys only see the percentages you don't see the actual numbers. Okay. That's right. Alright, that's fine. I will tell you what the actual number that is I see that there's no way I can actually say it so I'll put it in the chat I got
15:59:18 six out of 13 but plus Larry that's seven out of 13 for strongly

for so the gradation is strongly for weekly for neutral strongly against
sorry weekly against strongly against.
15:59:36 Okay.
15:59:37 and the number I got was 75200. And that's out of 13 people on the call. So that means how many people abstains.
15:59:52 That's up to 14.
15:59:54 Oh sorry that's out of 14 people yeah so everybody voted nobody abstained.
16:00:00 Well you'll have look Berg is here twice. Yes, I only wanted once
16:00:07 that Paul was taken. Yes. Yeah.
16:00:09 Okay. So, Nevertheless, this is a very strongly.
16:00:14 A strong indication that a lot of people like your paper.
16:00:19 Okay, it's pretty good for our first paper going good.
16:00:23 Thanks.
16:00:26 Well you tackled a really like graphics.
16:00:35 For instance what linear algebra does because we're not the only numerical algorithms in the world here right.
16:00:39 So what else, What else we got any exceptions.
16:00:42 It's full up. If you get it. If you get that data, it's, it's on your head.
16:00:48 You know so so the standard will give you as much information as it possibly can about what the acceptable range of data, and what to expect it, you know we we really go the extra mile.
16:00:57 But if we throw exceptions, then you know that you're effectively saying, Do not use this library.
16:01:05 It's, you know, and I would be very sad if that happened.
16:01:10 Alright guys, I think we should stop but thank you, you guys. You guys have been great experimental subjects for all my little experiments.
16:01:23 All right, next week next week, I mean next month. We'll see you guys, same place. And I think we might be back to graphs, but looks like we're going to have put some, some reinforcement learning there and maybe this time we can do an update, but looks
16:01:38 like this, people will have.
16:01:41 Well, actually we could probably wait until the time slot for stats to review this paper, which will probably be the month after that so I'll get all the scheduling straight.

On Tue, May 11, 2021 at 11:04 AM Michael Wong <fraggamuffin_at_[hidden]> wrote:

> SG19 Machine Learning 2 hours. This session will focus on Reinforcement

> *Learning and a review of Stats feedback.*
>
> *Hi,*
>
> *Michael Wong is inviting you to a scheduled Zoom meeting.*
>
> *Topic: SG19 monthly Dec 2020-Feb 2021*
> *Time: 02:00 PM Eastern Time (US and Canada) 1800 UTC Stats*
> *Every month on the Second Thu,*
> *May 13, 2021 02:00 PM ET 1800 UTC Reinformaent Learning and Diff*
> *Calculus*
> *June 10, 2021 02:00 PM ET 1800 UTC Graph*
> *Jul 8, 2021 02:00 PM ET 1800 UTC Stats and Combinatorics*
>
>
> *Join from PC, Mac, Linux, iOS or Android:*
> *https://iso.zoom.us/j/93084591725?pwd=3DK3QxZjJlcnljaE13ZWU5cTILNkx0Zz09*
> *Password: 035530*
>
> *Or iPhone one-tap :*
> *US: +13017158592,,93084591725# or +13126266799,,93084591725#*
> *Or Telephone:*
> *Dial(for higher quality, dial a number based on your current location=*
>):
> *US: +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799 or +1*
> *408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782*
> *or 877 853 5247 (Toll Free)*
> *Meeting ID: 930 8459 1725*
> *Password: 035530*
> *International numbers available: https://iso.zoom.us/u/agewu4X97*
>
> *Or Skype for Business (Lync):*
> *https://iso.zoom.us/skype/93084591725*
>
> *Agenda:*
>
> *1. Opening and introductions*
>
> *The ISO Code of conduct:*
> *https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.pdf*
> *The IEC Code of Conduct:*

>
> *https://basecamp.iec.ch/download/iec-code-of-conduct-for-delegates-and-ex*=
perts/
>
> *ISO patent policy.*
>
> *https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Po*=
licy.htm?nodeid=3D6344764&vernum=3D-2
>
> *The WG21 Practices and Procedures and Code of Conduct:*
>
> *https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedu*=
res
>
> *1.1 Roll call of participants*
>
> *1.2 Adopt agenda*
>
> *1.3 Approve minutes from previous meeting, and approve publishing*
> *previously approved minutes to ISOCPP.org*
>
> *1.4 Action items from previous meetings*
>
> *2. Main issues (125 min)*
>
> *2.1 General logistics*
>
> *Meeting plan, focus on one paper per meeting but does not preclude other*
> *paper updates:*
>
> *May 13, 2021 02:00 PM ET 1800 UTC Reinformaent Learning and Diff*
> *Calculus*
> *June 10, 2021 02:00 PM ET 1800 UTC Graph*
> *Jul 8, 2021 02:00 PM ET 1800 UTC Stats and Combinatorics*
>
> *ISO meeting status*
>
> *future C++ Std meetings*
>
> *2.2 Paper reviews*
>

> *2.2.1: ML topics*
>
> *2.2.1.1 Graph Proposal Phil Ratsloff et al*
>
> *P1709R1: Graph Proposal for Machine Learning*
>
> *P1709R3:*
>
> *https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyY=*
dRy4dM/edit?usp=3Dsharing
>
>
> *https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1U=*
g_vkkE/edit?usp=3Dsharing
>
> *<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>*
>
> *<*
>
> *https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E=*
8PoGS0/edit?ts=3D5fff27cd#heading=3Dh.9ogkehmdmtel
> >
>
> *2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva*
>
> *Reinforcement Learning proposal:*
>
> *2.2.1.3 Differential Calculs:*
>
>
> *https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E=*
8PoGS0/edit?ts=3D5fff27cd#heading=3Dh.9ogkehmdmtel
>
> *2.2.1.4: Stats paper*
>
> *Current github*
>

> https://github.com/cplusplus/papers/issues/475
>
> https://github.com/cplusplus/papers/issues/979
>
> Stats review Richard Dosselman et al
>
> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p1708r4.pdf
>
> Feedback from Johan Lundberg and Oleksandr Korval
>
> https://isocpp.org/files/papers/D2376R0.pdf
>
>
>
> P1708R3: Math proposal for Machine Learning: 3rd review
>
> PXXXX: combinatorics: 1st Review
>
> > std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2
> > above is the stats paper that was reviewed in Prague
> > http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19
> >
> > Review Jolanta Polish feedback.
> > http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html
>
> 2.2.3 any other proposal for reviews?
>
> 2.3 Other Papers and proposals
>
> P1416R1: SG19 - Linear Algebra for Data Science and Machine Learning
>
> https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk7xKi=
kK4Yp8/edit#heading=3Dh.tj9hitg7dbtr
>
> P1415: Machine Learning Layered list
>
> https://docs.google.com/document/d/1eINFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5t=
LVSj64/edit#heading=3Dh.tj9hitg7dbtr
>
> 2.2.2 SG14 Linear Algebra progress:

> *Different layers of proposal*
>
> *https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0lj=*
dUAtSQ/edit
>
> *2.5 Future F2F meetings:*
>
> *2.6 future C++ Standard meetings:*
> *https://isocpp.org/std/meetings-and-participation/upcoming-meetings*
>
> *None*
>
> *3. Any other business*
>
> *New reflector*
>
> *http://lists.isocpp.org/mailman/listinfo.cgi/sg19*
>
> *Old Reflector*
> *https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19*
> *<https://groups.google.com/a/isocpp.org/forum/?fromgroups=3D#!forum/sg14>*
>
> *Code and proposal Staging area*
>
> *4. Review*
>
> *4.1 Review and approve resolutions and issues [e.g., changes to SG's*
> *working draft]*
>
> *4.2 Review action items (5 min)*
>
> *5. Closing process*
>
> *5.1 Establish next agenda*
>
> *TBD*
>
> *5.2 Future meeting*
>
> *June 10, 2021 02:00 PM ET 1800 UTC Graph*

> *Jul 8, 2021 02:00 PM ET 1800 UTC Stats and Combinatorics*

Minutes for 2021/06/10 SG19 Conference Call

On Thu, Jun 10, 2021 at 12:18 PM will wray <wjwray_at_[hidden]> wrote:

P2128
variadic subscript operator takes more indexes instead of function call
operator overload
the authors felt it didnt matter either way

python compatible interface
it doesnt really fit within C/C++ type system

vector bool and valarray gives back proxy, so what is a comma separated
list in [ ] mean? rust and D is slice and returns array type
single index drops rank

selecting the indexes for that result
multidim array: switch from functional indexing to square brackets

mixed concerns:
n number papers are wg14 papers, this needs to be P number for WG21 excecpt
SG22
current paper only means we want mdspan to provide comma separated multi
dim, only allow operator [ ] to take comma separated only if you index a
class type, not builtin
so core language proposal is language agnostic
this is not different from operator () already do as there is no
inheritance

that is covered in P2128

the syntax shown here is new syntax

a..b is different then 1,2

conflict is expectation; if you index with a range, it is syntax error,

we do have invokable concept in C++20, for call operator, discussion on integers but we can index on more then integers
then why ask for variaidic? its expensive because of multiple objects at compile time, brace_init version has init list
its a popular feature but what is future costs

we want uniform syntax,

need to talk to more people about the cost to oppose a,b

there is a core lang feature to allow this syntax does not affetc builtin types to leave C do what they want in this area

objection mdspan using bracket multidim syntax, should wait for LEWG discussing mdspan, and that has nothing to do with C
no impact core on builti nunless non-member operators

>
>
> *On Wed, Jun 9, 2021 at 11:04 AM Michael Wong <fraggamuffin_at_[hidden]>*
> *wrote:*
>
>> *Thank you for the correction WIll. We should have time for your paper.*
>> *Thank you.*
>>
>> *On Wed, Jun 9, 2021 at 10:52 AM will wray <wjwray_at_[hidden]> wrote:*
>>
>>> *Should be tomorrow June 10, right? (Title says "June 9")*

>>>
>>> *The agenda looks quiet.*
>>> *If there is interest I'd like to give a short presentation, ~10 mins,*
>>> *with arguments*
>>> *against C++ proposal P2128 "Multidimensional subscript operator".*
>>> *I'm writing a rebuttal paper for SG22 and could do with feedback before*
>>> *submitting it.*
>>> *The topic is relevant to SG19 too, as it targets the mdspan API.*
>>> *I'll post a link here.*
>>>
>>> *On Wed, Jun 9, 2021 at 10:27 AM Michael Wong via SG19 <*
>>> *sg19_at_[hidden]> wrote:*
>>>
>>>> *SG19 Machine Learning 2 hours. This session will focus on Reinforcement*
>>>> *Learning and a review of Stats feedback from*
>>>> *P2376R0*
>>>> *<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2376r0.pdf>*
Comments
>>>> *on Simple Statistical Functions (p1708r4): Contracts, Exceptions and*
>>>> *Special cases Johan Lundberg*
>>>>
>>>> *Hi,*
>>>>
>>>> *Michael Wong is inviting you to a scheduled Zoom meeting.*
>>>>
>>>> *Topic: SG19 monthly*
>>>> *Time: 02:00 PM Eastern Time (US and Canada) 1800 UTC Stats*
>>>> *Every month on the Second Thu,*
>>>> *June 10, 2021 02:00 PM ET 1800 UTC Reinforcement Learning, stats*
>>>> *feedback*
>>>> *Jul 8, 2021 02:00 PM ET 1800 UTC Graph?*
>>>>
>>>> *Aug 12 2021 02:00 PM ET 1800 UTC Differential calculus*
>>>>
>>>> *Join from PC, Mac, Linux, iOS or Android:*
>>>> *https://iso.zoom.us/j/93084591725?pwd=K3QxZjJlcnljaE13ZWU5cTlLNkx0Zz0
9*
>>>> *Password: 035530*
>>>>
>>>> *Or iPhone one-tap :*
>>>> *US: +13017158592,,93084591725# or +13126266799,,93084591725#*

>>>> *Or Telephone:*
>>>> *Dial(for higher quality, dial a number based on your current*
>>>> *location):*
>>>> *US: +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799 or +1*
>>>> *408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782*
>>>> *or 877 853 5247 (Toll Free)*
>>>> *Meeting ID: 930 8459 1725*
>>>> *Password: 035530*
>>>> *International numbers available: https://iso.zoom.us/u/agewu4X97*
>>>>
>>>> *Or Skype for Business (Lync):*
>>>> *https://iso.zoom.us/skype/93084591725*
>>>>
>>>> *Agenda:*
>>>>
>>>> *1. Opening and introductions*
>>>>
>>>> *The ISO Code of conduct:*
>>>> *https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.pdf*
>>>> *The IEC Code of Conduct:*
>>>>
>>>> *https://basecamp.iec.ch/download/iec-code-of-conduct-for-delegates-and-experts/*
>>>>
>>>> *ISO patent policy.*
>>>>
>>>> *https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Policy.htm?nodeid=6344764&vernum=-2*
>>>>
>>>> *The WG21 Practices and Procedures and Code of Conduct:*
>>>>
>>>> *https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures*
>>>>
>>>> *1.1 Roll call of participants*
>>>>
>>>> *1.2 Adopt agenda*
>>>>
>>>> *1.3 Approve minutes from previous meeting, and approve publishing*
>>>> *previously approved minutes to ISOCPP.org*
>>>>
>>>> *1.4 Action items from previous meetings*

>>>>
>>>> 2. Main issues (125 min)
>>>>
>>>> 2.1 General logistics
>>>>
>>>> Meeting plan, focus on one paper per meeting but does not preclude other
>>>> paper updates:
>>>>
>>>> June 10, 2021 02:00 PM ET 1800 UTC Reinforcement Learning, stats
>>>> feedback
>>>>
>>>> Jul 8, 2021 02:00 PM ET 1800 UTC Graph?
>>>>
>>>> Aug 12 2021 02:00 PM ET 1800 UTC Differential calculus
>>>>
>>>> ISO meeting status
>>>>
>>>> future C++ Std meetings
>>>>
>>>> 2.2 Paper reviews
>>>>
>>>> 2.2.1: ML topics
>>>>
>>>> 2.2.1.1 Graph Proposal Phil Ratsloff et al
>>>>
>>>> P1709R1: Graph Proposal for Machine Learning
>>>>
>>>> P1709R3:
>>>>
>>>> https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing
>>>>
>>>>
>>>> https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing
>>>>
>>>> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>
>>>>
>>>> <
>>>>
>>>> https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjm

npfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel

>>>> >

>>>>

>>>> *Stats feedback:*

>>>>

>>>> *P2376R0*

>>>> <*http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2376r0.pdf*> *Comments*

>>>> *on Simple Statistical Functions (p1708r4): Contracts, Exceptions and*

>>>> *Special cases Johan Lundberg*

>>>>

>>>> *2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva*

>>>>

>>>> *Reinforcement Learning proposal:*

>>>>

>>>> *2.2.1.3 Differential Calculus:*

>>>>

>>>>

>>>> *https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjm npfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel*

>>>>

>>>> *2.2.1.4: Stats paper*

>>>>

>>>> *Current github*

>>>>

>>>> *https://github.com/cplusplus/papers/issues/475*

>>>>

>>>> *https://github.com/cplusplus/papers/issues/979*

>>>>

>>>> *Stats review Richard Dosselman et al*

>>>>

>>>> *http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p1708r4.pdf*

>>>>

>>>> *Feedback from Johan Lundberg and Oleksandr Korval*

>>>>

>>>> *https://isocpp.org/files/papers/D2376R0.pdf*

>>>>

>>>> *P1708R3: Math proposal for Machine Learning: 3rd review*

>>>>

>>> *Oleksandr feedback:*

mode should be included

python just does blind sort, mlogm

conflict on some formulas
mean, skewness are fine
kurtosis too
pearson variant deletes the second term
swap order of parameter in Clause 4

4.1.1 call the mean without projection, on a std::vector
if vector <float> means we get a float, not double because integral, if
int, then I will get double back
what is customization anticipated, please use long double?

do we want a second one? No i want something else
2 problems: can only specify templ argument positionally R, P first before
I can give value for capital result parameter
this is unfriendly, depends on Rp
2. general algo front matter say must specify templ arg explicitly because
order can change between impl

have we looked at non commutativity? no, we dont say anything order with
parallel

can we use projection that converts to long double? third parameter is
fully defined by the other 2 so yes

in 4.1: why 2 kind of return parameter for integer and float? why not just
float? if any integral, then double, double becomes double, float becomes
float
can integer become float?
integer vector then makes no sense to compute the mean
embedded HW where floats are done in HW and double is emulated, so taking
result of projection customized may be right
yes GPUs can have some bigger number of floats

prefer we go with simple option returning result from deduction, so want
promoted integral type? for mean ok but std. deviation

if you keep natural return type of operation, then you overflow, so agree
with this doube conversion by default for integral types because it gives
the correct result

yes integers are problematic case and likely overflow as Marco said, in my environment

want easy way for people to specify projection

LEWG have 2 months notice

Johan ask that empty range should return naN

this part needs rationale, and a period at the end of sentences, + infinity with mean, why not infinity? 0+infinity compute mean is not infinity. Yes but the mathematical definition of mean gets infinity by IEEE, inifnity/2=infinity

any deviation from IEEE mathematical thing needs to be explained, not opposed

on balance, prefer what the naive impl is if the behaviour of algo follows that, helps to reduce surprise, especially if input is not useful like inf, then possibly a branch to reduce efficiency, should be optimized for sensible inputs, least surprise

yes, I find this surprising and agrees with Jens, especially if both sequences empty, inf does not make sense,

integral or integer type returned without infinity, like fixed point type in particular, also rational types as well

since this is a required fn parameter, it does not need a default

adding a virtual fn base abstract interface is no to me, due to performance, inlining issue

customization in C++ we just add another template parameter or use a concept of what an accumulator is, also why the copy assignment operator

another is parallel accumulation without synchronization, and have an additional method like aggregate

custom accumulator object is OK? yes

stddev or spelled out in full should be asked to LEWG as a question and justified

might not matter that it is like random number

not having range here can change the interafce, leave it as open issue

a division involved here, divide by projected type, in general floating point accumulation is complex, so we dont have raw doubles
a projection should interfere with the type as little as possible; can auto deduce the return type of this by default

naming Fisher Pearson from mumpy is it better to have descriptive names, so Fisher may be the only type; no real standard

also might want to limit the type of these somehwo; can it be any random type, or same type as the projected type, ok ignore me here

can it be arithmetic else illformed? then that constraint cannot be

expanded? NO user cannot put overloads in std namespace

PXXXX: combinatorics: 1st Review

>>>> > *Review Jolanta Polish feedback.*
>>>> > *http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html*
>>>>
>>>> *2.2.3 any other proposal for reviews?*
>>>>
>>>> *2.3 Other Papers and proposals*
>>>>
>>>> *P1416R1: SG19 - Linear Algebra for Data Science and Machine Learning*
>>>>
>>>> *https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk7xKikK4Yp8/edit#heading=h.tj9hitg7dbtr*
>>>>
>>>> *P1415: Machine Learning Layered list*
>>>>
>>>> *https://docs.google.com/document/d/1eINFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr*
>>>>
>>>> *2.2.2 SG14 Linear Algebra progress:*
>>>> *Different layers of proposal*
>>>>
>>>> *https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXIK_d0IjdUAtSQ/edit*
>>>>
>>>> *2.5 Future F2F meetings:*
>>>>
>>>> *2.6 future C++ Standard meetings:*
>>>> *https://isocpp.org/std/meetings-and-participation/upcoming-meetings*
>>>>
>>>> *None*
>>>>
>>>> *3. Any other business*
>>>>
>>>> *New reflector*
>>>>
>>>> *http://lists.isocpp.org/mailman/listinfo.cgi/sg19*
>>>>
>>>> *Old Reflector*
>>>> *https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19*
>>>> *<https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/sg14>*
>>>>
>>>> *Code and proposal Staging area*

>>>>
>>>> 4. Review
>>>>
>>>> 4.1 Review and approve resolutions and issues [e.g., changes to SG's
>>>> working draft]
>>>>
>>>> 4.2 Review action items (5 min)
>>>>
>>>> 5. Closing process
>>>>
>>>> 5.1 Establish next agenda
>>>>
>>>> TBD
>>>>
>>>> 5.2 Future meeting
>>>>
>>>>
>>>> Jul 8, 2021 02:00 PM ET 1800 UTC Graph?
>>>>
>>>> Aug 12 2021 02:00 PM ET 1800 UTC Differential calculus
>>>> --
>>>> SG19 mailing list
>>>> SG19_at_[hidden]
>>>> https://lists.isocpp.org/mailman/listinfo.cgi/sg19

Minutes for 2021/07/08 SG19 Conference Call

Here are the attendees:
Michael Wong
Richard Dosselmann
Oleksandr Koval
Jens Maurer
Guy Davidson
Marco Foco
John McFarlane
Will Wray
Andrew Lumsdaine
Ren=C3=A9 Ferdinand Rivera Morell
Oleksandr Koval
Ozan Irsoy
Scott Moe
Phil Ratzloff
Ritwik Dubey
Ronan Keryell
Luke D'Alessandro
Scott Moe
Sayan Ghosh (PNNL)
Izzy Muerte

On Thu, Jun 10, 2021 at 3:59 PM Michael Wong <fraggamuffin_at_[hidden]> wrote=
:

>
>
> *On Thu, Jun 10, 2021 at 12:18 PM will wray <wjwray_at_[hidden]> wrote:*
>
>> *Here's a link to the P2128 rebuttal draft:*
>> **n2740* Arguments against a[i...] syntax for multidimensional indexing*
>> *<https://github.com/willwray/n2740/blob/main/P2128_rebuttal.md>*
>> *A bit long - here's a direct link to the Case Summary*
>> *<https://github.com/willwray/n2740/blob/main/P2128_rebuttal.md#case-summ=*
ary>
>>

>>
>
> P2128
> variadic subscript operator takes more indexes instead of function call
> operator overload
> the authors felt it didnt matter either way
>
> python compatible interface
> it doesnt really fit within C/C++ type system
>
> vector bool and valarray gives back proxy, so what is a comma separated
> list in [ ] mean? rust and D is slice and returns array type
> single index drops rank
>
> selecting the indexes for that result
> multidim array: switch from functional indexing to square brackets
>
>
>
> mixed concerns:
> n number papers are wg14 papers, this needs to be P number for WG21
> excecpt SG22
> current paper only means we want mdspan to provide comma separated multi
> dim, only allow operator [ ] to take comma separated only if you index a
> class type, not builtin
> so core language proposal is language agnostic
> this is not different from operator () already do as there is no
> inheritance
>
> that is covered in P2128
>
> the syntax shown here is new syntax
>
> a..b is different then 1,2
>
> conflict is expectation; if you index with a range, it is syntax error,
>
>

> we do have invokable concept in C++20, for call operator, discussion on
> integers but we can index on more then integers
> then why ask for variaidic? its expensive because of multiple objects at
> compile time, brace_init version has init list
> its a popular feature but what is future costs
>
> we want uniform syntax,
>
>
> need to talk to more people about the cost to oppose a,b
>
>
> there is a core lang feature to allow this syntax does not affetc builtin
> types to leave C do what they want in this area
>
> objection mdspan using bracket multidim syntax, should wait for LEWG
> discussing mdspan, and that has nothing to do with C
> no impact core on builti nunless non-member operators
>
>
>
>
>
>
>
>
>>
>>
>> On Wed, Jun 9, 2021 at 11:04 AM Michael Wong <fraggamuffin_at_[hidden]>
>> wrote:
>>
>>> Thank you for the correction WIll. We should have time for your paper.
>>> Thank you.
>>>
>>> On Wed, Jun 9, 2021 at 10:52 AM will wray <wjwray_at_[hidden]> wrote:
>>>
>>>> Should be tomorrow June 10, right? (Title says "June 9")
>>>>

>>>> *The agenda looks quiet.*
>>>> *If there is interest I'd like to give a short presentation, ~10 mins,*
>>>> *with arguments*
>>>> *against C++ proposal P2128 "Multidimensional subscript operator".*
>>>> *I'm writing a rebuttal paper for SG22 and could do with feedback befor=*
e
>>>> *submitting it.*
>>>> *The topic is relevant to SG19 too, as it targets the mdspan API.*
>>>> *I'll post a link here.*
>>>>
>>>> *On Wed, Jun 9, 2021 at 10:27 AM Michael Wong via SG19 <*
>>>> *sg19_at_[hidden]> wrote:*
>>>>
>>>>> *SG19 Machine Learning 2 hours. This session will focus on Reinforceme=*
nt
>>>>> *Learning and a review of Stats feedback from*
>>>>> *P2376R0*
>>>>> *<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2376r0.pdf>=*
 Comments
>>>>> *on Simple Statistical Functions (p1708r4): Contracts, Exceptions and*
>>>>> *Special cases Johan Lundberg*
>>>>>
>>>>> *Hi,*
>>>>>
>>>>> *Michael Wong is inviting you to a scheduled Zoom meeting.*
>>>>>
>>>>> *Topic: SG19 monthly*
>>>>> *Time: 02:00 PM Eastern Time (US and Canada) 1800 UTC Stats*
>>>>> *Every month on the Second Thu,*
>>>>> *June 10, 2021 02:00 PM ET 1800 UTC Reinforcement Learning, stats*
>>>>> *feedback*
>>>>> *Jul 8, 2021 02:00 PM ET 1800 UTC Graph?*
>>>>>
>>>>> *Aug 12 2021 02:00 PM ET 1800 UTC Differential calculus*
>>>>>
>>>>> *Join from PC, Mac, Linux, iOS or Android:*
>>>>> *https://iso.zoom.us/j/93084591725?pwd=3DK3QxZjJlcnljaE13ZWU5cTlLNkx0*
Z=

z09

>>>>> *Agenda:*

>>>>>

>>>>> *1. Opening and introductions*

>>>>>

>>>>> *The ISO Code of conduct:*

>>>>>

>>>>> *https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.=*

pdf

>>>>> *The IEC Code of Conduct:*

>>>>>

>>>>> *https://basecamp.iec.ch/download/iec-code-of-conduct-for-delegates-an=*

d-experts/

>>>>>

>>>>> *ISO patent policy.*

>>>>>

>>>>> *https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Commo=*

n_Policy.htm?nodeid=3D6344764&vernum=3D-2

>>>>>

>>>>> *The WG21 Practices and Procedures and Code of Conduct:*

>>>>>
>>>>> https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-pro=
cedures
>>>>>
>>>>> *1.1 Roll call of participants*
>>>>>
>>>>> *1.2 Adopt agenda*
>>>>>
>>>>> *1.3 Approve minutes from previous meeting, and approve publishing*
>>>>> *previously approved minutes to ISOCPP.org*
>>>>>
>>>>> *1.4 Action items from previous meetings*
>>>>>
>>>>> *2. Main issues (125 min)*
>>>>>
>>>>> *2.1 General logistics*
>>>>>
>>>>> *Meeting plan, focus on one paper per meeting but does not preclude*
>>>>> *other*
>>>>> *paper updates:*
>>>>>
>>>>> *June 10, 2021 02:00 PM ET 1800 UTC Reinforcement Learning, stats*
>>>>> *feedback*
>>>>>
>>>>> *Jul 8, 2021 02:00 PM ET 1800 UTC Graph?*
>>>>>
>>>>> *Aug 12 2021 02:00 PM ET 1800 UTC Differential calculus*
>>>>>
>>>>> *ISO meeting status*
>>>>>
>>>>> *future C++ Std meetings*
>>>>>
>>>>> *2.2 Paper reviews*
>>>>>
>>>>> *2.2.1: ML topics*
>>>>>
>>>>> *2.2.1.1 Graph Proposal Phil Ratsloff et al*
>>>>>

>>>>> *P1709R1: Graph Proposal for Machine Learning*

>>>>>

>>>>> *P1709R3:*

>>>>>

>>>>> [*https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5*](https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5)=

_dyYdRy4dM/edit?usp=3Dsharing

>>>>>

>>>>>

>>>>> [*https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJg*](https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJg)=

zg1Ug_vkkE/edit?usp=3Dsharing

>>>>>

>>>>> <[*http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html*](http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html)=

>

>>>>>

>>>>> <

>>>>>

>>>>> [*https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpf*](https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpf)=

c-_E8PoGS0/edit?ts=3D5fff27cd#heading=3Dh.9ogkehmdmtel

>>>>> >

>>>>>

>>>>> *Stats feedback:*

>>>>>

>>>>> *P2376R0*

>>>>> <[*http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2376r0.pdf*](http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2376r0.pdf)>= Comments

>>>>> *on Simple Statistical Functions (p1708r4): Contracts, Exceptions and*

>>>>> *Special cases Johan Lundberg*

>>>>>

>>>>> *2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva*

>>>>>

>>>>> *Reinforcement Learning proposal:*

>>>>>

>>>>> *2.2.1.3 Differential Calculus:*

>>>>>

>>>>>

>>>>> [https://docs.google.com/document/d/175wlm8o4BNGti0WLq8U6uZORegKVj mnpf](https://docs.google.com/document/d/175wlm8o4BNGti0WLq8U6uZORegKVj)=
c-_E8PoGS0/edit?ts=3D5fff27cd#heading=3Dh.9ogkehmdmtel

>>>>>

>>>>> 2.2.1.4: Stats paper

>>>>>

>>>>> Current github

>>>>>

>>>>> https://github.com/cplusplus/papers/issues/475

>>>>>

>>>>> https://github.com/cplusplus/papers/issues/979

>>>>>

>>>>> Stats review Richard Dosselman et al

>>>>>

>>>>> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p1708r4.pdf

>>>>>

>>>>> Feedback from Johan Lundberg and Oleksandr Korval

>>>>>

>>>>> https://isocpp.org/files/papers/D2376R0.pdf

>>>>>

>>>>> P1708R3: Math proposal for Machine Learning: 3rd review

>>>>>

>>>> Oleksandr feedback:

> mode should be included

> python just does blind sort, mlogm

>

> conflict on some formulas

> mean, skewness are fine

> kurtosis too

> pearson variant deletes the second term

> swap order of parameter in Clause 4

>

> 4.1.1 call the mean without projection, on a std::vector

> if vector <float> means we get a float, not double because integral, if

> int, then I will get double back

> what is customization anticipated, please use long double?

>

> do we want a second one? No i want something else

> *2 problems: can only specify templ argument positionally R, P first befor=*
e
> *I can give value for capital result parameter*
> *this is unfriendly, depends on Rp*
> *2. general algo front matter say must specify templ arg explicitly becaus=*
e
> *order can change between impl*
>
> *have we looked at non commutativity? no, we dont say anything order with*
> *parallel*
>
> *can we use projection that converts to long double? third parameter is*
> *fully defined by the other 2 so yes*
>
> *in 4.1: why 2 kind of return parameter for integer and float? why not jus=*
t
> *float? if any integral, then double, double becomes double, float becomes*
> *float*
> *can integer become float?*
> *integer vector then makes no sense to compute the mean*
> *embedded HW where floats are done in HW and double is emulated, so taking*
> *result of projection customized may be right*
> *yes GPUs can have some bigger number of floats*
>
> *prefer we go with simple option returning result from deduction, so want*
> *promoted integral type? for mean ok but std. deviation*
>
> *if you keep natural return type of operation, then you overflow, so agree*
> *with this doube conversion by default for integral types because it give=*
s
> *the correct result*
>
> *yes integers are problematic case and likely overflow as Marco said, in m=*
y
> *environment*
>
> *want easy way for people to specify projection*
>

> LEWG have 2 months notice
>
> Johan ask that empty range should return naN
>
> this part needs rationale, and a period at the end of sentences, +
> infinity with mean, why not infinity? 0+infinity compute mean is not
> infinity. Yes but the mathematical definition of mean gets infinity by
> IEEE, inifnity/2=3Dinfinity
>
> any deviation from IEEE mathematical thing needs to be explained, not
> opposed
>
> on balance, prefer what the naive impl is if the behaviour of algo follow=
s
> that, helps to reduce surprise, especially if input is not useful like in=
f,
> then possibly a branch to reduce efficiency, should be optimized for
> sensible inputs, least surprise
>
> yes, I find this surprising and agrees with Jens, especially if both
> sequences empty, inf does not make sense,
>
> integral or integer type returned without infinity, like fixed point type
> in particular, also rational types as well
>
>
> since this is a required fn parameter, it does not need a default
>
> adding a virtual fn base abstract interface is no to me, due to
> performance, inlining issue
>
> customization in C++ we just add another template parameter or use a
> concept of what an accumulator is, also why the copy assignment operator
>
> another is parallel accumulation without synchronization, and have an
> additional method like aggregate
>
> custom accumulator object is OK? yes

>
> *stddev or spelled out in full should be asked to LEWG as a question and*
> *justified*
>
> *might not matter that it is like random number*
>
> *not having range here can change the interafce, leave it as open issue*
>
> *a division involved here, divide by projected type, in general floating*
> *point accumulation is complex, so we dont have raw doubles*
> *a projection should interfere with the type as little as possible; can*
> *auto deduce the return type of this by default*
>
>
> *naming Fisher Pearson from mumpy is it better to have descriptive names,*
> *so Fisher may be the only type; no real standard*
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
> *also might want to limit the type of these somehwo; can it be any random*
> *type, or same type as the projected type, ok ignore me here*

>
> *can it be arithmetic else illformed? then that constraint cannot be*
> *expanded? NO user cannot put overloads in std namespace*
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
> *PXXXX: combinatorics: 1st Review*

>>>>>

>>>>> > std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2

>>>>> > above is the stats paper that was reviewed in Prague

>>>>> > http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19

>>>>> >

>>>>> > Review Jolanta Polish feedback.

>>>>> > http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.htm=
l

>>>>>

>>>>> 2.2.3 any other proposal for reviews?

>>>>>

>>>>> 2.3 Other Papers and proposals

>>>>>

>>>>> P1416R1: SG19 - Linear Algebra for Data Science and Machine Learning

>>>>>

>>>>> https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk=
7xKikK4Yp8/edit#heading=3Dh.tj9hitg7dbtr

>>>>>

>>>>> P1415: Machine Learning Layered list

>>>>>

>>>>> https://docs.google.com/document/d/1eINFdIXWoetbxjO1OKol_Wj8fyi4Z4hog=
fj5tLVSj64/edit#heading=3Dh.tj9hitg7dbtr

>>>>>

>>>>> 2.2.2 SG14 Linear Algebra progress:

>>>>> Different layers of proposal

>>>>>

>>>>> https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXIK_=
d0IjdUAtSQ/edit

>>>>>

>>>>> 2.5 Future F2F meetings:

>>>>>

>>>>> 2.6 future C++ Standard meetings:

>>>>> https://isocpp.org/std/meetings-and-participation/upcoming-meetings

>>>>>

>>>>> None

>>>>>
>>>>> *3. Any other business*
>>>>>
>>>>> *New reflector*
>>>>>
>>>>> *http://lists.isocpp.org/mailman/listinfo.cgi/sg19*
>>>>>
>>>>> *Old Reflector*
>>>>> *https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19*
>>>>> *<https://groups.google.com/a/isocpp.org/forum/?fromgroups=3D#!forum/s*=
g14
>>>>> >
>>>>>
>>>>> *Code and proposal Staging area*
>>>>>
>>>>> *4. Review*
>>>>>
>>>>> *4.1 Review and approve resolutions and issues [e.g., changes to SG's*
>>>>> *working draft]*
>>>>>
>>>>> *4.2 Review action items (5 min)*
>>>>>
>>>>> *5. Closing process*
>>>>>
>>>>> *5.1 Establish next agenda*
>>>>>
>>>>> *TBD*
>>>>>
>>>>> *5.2 Future meeting*
>>>>>
>>>>>
>>>>> *Jul 8, 2021 02:00 PM ET 1800 UTC Graph?*
>>>>>
>>>>> *Aug 12 2021 02:00 PM ET 1800 UTC Differential calculus*
>>>>> *--*
>>>>> *SG19 mailing list*
>>>>> *SG19_at_[hidden]*
>>>>> *https://lists.isocpp.org/mailman/listinfo.cgi/sg19*

Minutes for 2021/09/09 SG19 Conference Call

> > Hi all, SG19 Sept meeting is tomorrow Sept 9.

> >

> > SG19 Machine Learning 2 hours. This session will focus on Graph

> >

> > Hi,

> >

> > Michael Wong is inviting you to a scheduled Zoom meeting.

> >

> > Topic: SG19 monthly

> > Time: 02:00 PM Eastern Time (US and Canada) 1800 UTC Stats

> > Every month on the Second Thu,

> >

> >

> > Join from PC, Mac, Linux, iOS or Android:

> >

> > https://iso.zoom.us/j/93084591725?pwd=K3QxZjJlcnljaE13ZWU5cTlLNkx0Zz09

> > Password: 035530

> >

> > Or iPhone one-tap :

> > US: +13017158592,,93084591725# or +13126266799,,93084591725#

> > Or Telephone:

> > Dial(for higher quality, dial a number based on your current

> location):

> > US: +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799 or +1

> > 408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782

> > or 877 853 5247 (Toll Free)

> > Meeting ID: 930 8459 1725

> > Password: 035530

> > International numbers available: https://iso.zoom.us/u/agewu4X97

> >

> > Or Skype for Business (Lync):

> > https://iso.zoom.us/skype/93084591725

> >

> > Agenda:

> >

> > 1. Opening and introductions

> >

> > The ISO Code of conduct:

> > https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.pdf

> >

> > IEC Code of Conduct:

> >

> > https://www.iec.ch/basecamp/iec-code-conduct-technical-work

> >

> > ISO patent policy.

> >

> >

> https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Policy.htm?nodeid=6344764&vernum=-2

> >

> > The WG21 Practices and Procedures and Code of Conduct:

> >

> https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures

> >

> > 1.1 Roll call of participants

> >

> > 1.2 Adopt agenda

> >

> > 1.3 Approve minutes from previous meeting, and approve publishing

> > previously approved minutes to ISOCPP.org

> >

> > 1.4 Action items from previous meetings

> >

> > 2. Main issues (125 min)

> >

> > 2.1 General logistics

> >

> > Meeting plan, focus on one paper per meeting but does not preclude other

> > paper updates:

> >

> > Sep 9, 2021 02:00 PM ET 1800 UTC Graph

> >

> > Oct 14, 2021 02:00 PM ET 1800 UTC Differential calculus/Reinforcement

> > Learning

> >

> > Nov 11 , 2021 02:00 PM ET 1800 UTC DST break/cancelled

> >

> > Dec 9 2021 02:00 PM ET 1800 UTC Stats

> >

> > ISO meeting status

> >

> > *future C++ Std meetings*

> >

> > *2.2 Paper reviews*

> >

> > *2.2.1: ML topics*

> >

> > *2.2.1.1 Graph Proposal Phil Ratsloff et al*

> >

> > *Latest paper:*

> >

> > *Here?s a link to the paper (different than the previous paper reviewed).*
> > *There are some additional updates I?m planning on making before the*
> *meeting.*

> >

> >

> https://docs.google.com/document/d/1OpH-xxRri7tJTtJJIZTYmSHkkrZJkdBwm9zJ7LgoIfQ/edit?usp=sharing

> >

> >

>

Andrew Lumsdaine' proposal.

> >

> >

> > *P1709R3:*

> >

> https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing

> >

> >

> https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing

> >

> > <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>

> >

> > <

> >

> https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel

> >>

> >

> > *Array copy semantics:*
> > *array copy-semantics paper P1997 "Relaxing Restrictions on Arrays",*
> > *https://wg21.link/p1997*
> >
> > *Stats feedback:*
> >
> > *P2376R0*
> > *<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2376r0.pdf>*
> > *Comments*
> > *on Simple Statistical Functions (p1708r4): Contracts, Exceptions and*
> > *Special cases Johan Lundberg*
> >
> > *2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva*
> >
> > *Reinforcement Learning proposal:*
> >
> > *2.2.1.3 Differential Calculus:*
> >
> >
> *https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel*
> >
> > *2.2.1.4: Stats paper*
> >
> > *Current github*
> >
> > *https://github.com/cplusplus/papers/issues/475*
> >
> > *https://github.com/cplusplus/papers/issues/979*
> >
> > *Stats review Richard Dosselman et al*
> >
> > *http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p1708r4.pdf*
> >
> > *Feedback from Johan Lundberg and Oleksandr Korval*
> >
> > *https://isocpp.org/files/papers/D2376R0.pdf*
> >
> > *P1708R3: Math proposal for Machine Learning: 3rd review*
> >
> > *PXXXX: combinatorics: 1st Review*

> >
> >> std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2
> >> above is the stats paper that was reviewed in Prague
> >> http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19
> >>
> >> Review Jolanta Polish feedback.
> >> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html
> >
> > 2.2.3 any other proposal for reviews?
> >
> > 2.3 Other Papers and proposals
> >
> > P1416R1: SG19 - Linear Algebra for Data Science and Machine Learning
> >
> https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk7xKikK4Yp8/edit#heading=h.tj9hitg7dbtr
> >
> > P1415: Machine Learning Layered list
> >
> https://docs.google.com/document/d/1elNFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr
> >
> > 2.2.2 SG14 Linear Algebra progress:
> > Different layers of proposal
> >
> https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXIK_d0IjdUAtSQ/edit
> >
> > 2.5 Future F2F meetings:
> >
> > 2.6 future C++ Standard meetings:
> > https://isocpp.org/std/meetings-and-participation/upcoming-meetings
> >
> > None
> >
> > 3. Any other business
> >
> > New reflector
> >
> > http://lists.isocpp.org/mailman/listinfo.cgi/sg19
> >

> > Old Reflector
> > https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19
> > <https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/sg14>
> >
> > Code and proposal Staging area
> >
> > 4. Review
> >
> > 4.1 Review and approve resolutions and issues [e.g., changes to SG's
> > working draft]
> >
> > 4.2 Review action items (5 min)
> >
> > 5. Closing process
> >
> > 5.1 Establish next agenda
> >
> > Oct 14
> >
> > 5.2 Future meeting
> >
> >
> >
> > Oct 14, 2021 02:00 PM ET 1800 UTC Differential calculus/Reinforcement
> > Learning
> >
> > Nov 11 , 2021 02:00 PM ET 1800 UTC DST break/cancelled
> >
> > Dec 9 2021 02:00 PM ET 1800 UTC Stats
> >
>
>
>
> --
> This e-mail was sent via the Secure Web Server at the
> University of Regina, Department of Computer Science
> https://www.cs.uregina.ca/WebMail/
>
>
>
> --

--000000000000ef598905cbe191f7 Content-Type: text/plain; charset="UTF-8"

HI all, here is the closed caption notes:

14:03:01 Hi everybody, welcome to the September monthly SG 19th call. I'm just setting up.
14:03:22 So on the call.
14:03:23 We have people are just joining but since zoom is pick up all the attendees automatically just going to call out a few names so I have Andrew lumps thing.
14:03:34 Benjamin Brock, Yulia Birla Larry Lewis Luke della Sandro also no soy, Phil rats la Renee Ferdinand revere morale. cn go to see the rest of the name.
14:03:53 Guess saying Gosh, Scott McMillan Scott Moore will Ray and Joe sacks and I'm like a wall of SG 19.
14:04:05 So just moving through our agenda.
14:04:13 We have.
14:04:16 So that's the roll call the agenda today, I believe, is to go over two proposals the main source is the graph proposal I think we've been reviewing it as we're going along, and there's been some changes.
14:04:30 And I believe.
14:04:32 Phil has something that he's ready to show us he's excited about showing the current progress. It's already gone over, it's been reviewed pretty much almost all of for over a year now actually almost two years since the pandemic.
14:04:46 And so, there's been a lot of modifications, and it's, it's been going on long nicely so I think this is just another intermediate checkpoint, to see where we are, I believe, Andrew has something that he's prepared.
14:05:00 And there is a and will I think you have you have a proxy for a, an array proposal that you want to consider.
14:05:11 That's right, it's more of a news update, so it's something that's ongoing, for the last month. Right. And it's just one day this machine learning group on the progress of that.
14:05:21 Ray, is it is the author here for that or you know i i sent out messages to the two authors and receive a reply is but I don't think I don't think we need that it's a small modest proposal.

14:05:33 Okay, that's fine.

14:05:38 Any other issues people want to bring up. So what we do what we do is we rotate between three proposals three or four proposals. And right now, because we took a break in August for the summer.

14:05:52 September is the graph proposal the October meeting is probably going to go back to calculus and reinforcement learning but it could be replaced by something else depending on whether people have other proposals they have in mind.

14:06:07 November because of the DST change we usually take a break and cancel, and that puts stats.

14:06:13 Back on the list on December 9.

14:06:17 So that's the current current plan.

14:06:21 We don't have the stats how to offer here today Richard Dawson is a professor. And I think we're China, and he's teaching right now so he can't. He can't come, but that's okay, because that's it would not going to be talking about stats today.

14:06:35 It's his proposal has actually progressed into LWG meeting at some, it's already outside it's already been loaded out of this group.

14:06:46 So module any kind of changes from LDL p wg, but we will leave it, leave it where it is. Okay. Does anyone have any other things that people want to talk about before I go into general.

14:06:55 So that's part of general logistics numbers section 2.1. And I'll talk a little bit about ISO meeting status.

14:07:03 Let me think there is a plenary in October for ISO c++ is a virtual plenary.

14:07:11 And then there's an attempt at potential face to face meeting. In February, in Portland, but because of the pandemic uncertainty and delta variants and people's quarantines and varying varying restrictions on travels from different countries to different

14:07:30 countries.

14:07:31 There's now talk of moving, that one also to virtual, but nothing has been decided yet they're still looking at either virtual hybrid or physical.

14:07:42 So, these this is just the sign of the times, his face.

14:07:46 Okay so that pretty much gets through all of my logistics things to other people have it any logistics, or any other things that they want to bring up before we launch into the papers.

14:07:59 Okay, well Hearing none, let's proceed, given that will raise.

14:08:12 Ray update is probably the shoulders. Would you say, well, we'll do you want to do the update right away and then we'll move into the grant proposal. And I think if that's, is that okay, that works for me and just

tell tell me when to stop.

14:08:20 Okay, I'm graph proposal, people

14:08:25 Phil and and Andrew, are you Do you guys have any deadlines that you need to to like get a drop off or something like that.

14:08:33 No, I'm fine.

14:08:35 Okay.

14:08:37 Andrew Are you wanting to where you plan on covering something outside of our paper. It was just the intro pieces of the paper.

14:08:46 Okay, so what I'd like to do on that is just cover the, the, the, the stuff that I've done, and then we hopefully we'll have some time at the end to look at the intros Are you okay with that.

14:09:01 Sure.

14:09:03 Okay. Okay.

14:09:05 Unless you want to do, intro first to lead into, you know, set up to what you were your stuff.

14:09:13 Either way.

14:09:15 Yeah, I

14:09:18 don't know how long I can see us taking 45 minutes or so on, on my stuff fence on the, the feedback.

14:09:29 Okay. An angel How long do you think your peaceful be

14:09:34 probably about the same. Okay.

14:09:44 That sounds fine yeah 1015 minutes, 10, talking, 5% feedback.

14:09:49 That sounds good. Okay, why don't why don't we start.

14:09:52 So let's see if I can find it.

14:09:57 Oh, there it is.

14:10:00 Okay, go ahead. Go ahead. Well, so yes I can just talk over the paper if you like I have some time so I could show, I didn't do any slides.

14:10:09 So, this proposal is a small proposal was put in about 20 months ago first in December 2019 and then quickly a first revision.

14:10:20 The authors are talking about a second revision. The main author, a young guy amazing guy he worked for a while, for the c++ alliance.

14:10:31 And somehow, you know he's still a student I think but got into standard the US reading the standard.

14:10:35 I'm not sure who the second author is he's perhaps an academic. But, anyway, the, the paper itself is somewhat academic it.

14:10:46 You know, it includes the wording.

14:10:48 It's been criticized by some people high up in the c++ Committee is lacking motivation.

14:10:54 I can talk to that if we need so the reason why I think this is of interest, the machine learning group is it's related, and the remit here is pretty much rate related when you read it.

14:11:07 Now, this is going back 50 years really fixing an issue from see that then was inherited into c++, that arrays are not regular types, you can't copy them to be fully regular you need to compare as well, a semi regular type is called people.

14:11:24 And that can mostly be fixed and I don't understand why it wasn't done back when Richie made struct comfortable. So for a while in early. See, you would have to copy every single member of a struct when you copy destruct but somewhere between one and

14:11:42 two, structural help people with your rep with a raised hand side.

14:11:44 So that's one place where you get to wake up in the C, and c++.

14:11:48 The other singular cases string literal Wars character array can be initialized from right hand side string literal.

14:12:02 Now, in recent years have been a couple of other cases of array copies have been added, where that is convenient. So, lenders will copy the by value capture of an array structure bindings will copy an array, right hand side, oh sorry I'm not talking over

14:12:22 the paper here at all. That's not included. So there is a nice examples in the paper. So one of the questions that comes up incessantly is, why isn't good enough.

14:12:34 And I don't see that as a relevant question I want good built in a race. I want them in order to build better class arrays. So, I've been kind of working in array related stuff for many years, I did blessing, implementation, many years ago.

14:12:50 I'm kind of in the Python data, ecosystem at the moment, used to do MATLAB

14:12:56 arrays, can be comfortable. I, you know, this paper hasn't seen any, any issues raised against it people haven't come up with any reasons why it won't work.

14:13:07 I've been trying to look at it from, you know, an attackers point of view what what could be wrong with this what doesn't work. I, I know of two small breakages that were discussed with this see.

14:13:22 Okay, so what's the history of this, so 20 months ago, it hasn't seen any interest. I'm the main proponent keep pushing on it, the author has gone on to other things, but it's still kind of keeping his hand in a little probably would be involved with

14:13:35 this, if there's a revision.

14:13:37 So about a year ago, I joined this group, mostly to push rate related things and the SG 22, the newly formed C and c++ liaison group, partly because I realized that they were more likely to be interested in this proposal then c++.

14:13:56 As you know, array is a somewhat to boo within the c++ Committee

and community where it's still very central to see.

14:14:03 I mean that's foundational in c++, as well, but it creates something of a two worlds view of and that you know, coming back to standard array, it's kind of typify by standard array.

14:14:16 So what's the problem with standard rate. One of the problems listed in this paper is the initialization issue. So I call this the member array initialization conundrum.

14:14:28 And there is the few kind of work around solutions I've blogged on a few of them.

14:14:35 But the problem is, you have an existing l value array. How do you initialize the standard away with it.

14:14:41 Now we have stood ranges copy. So ranges copy can do can do that for a one dimensional array, but fails, as soon as it is nested, so it won't do it for nested see array.

14:14:53 I believe it won't do it for nested standard right but I haven't tried that

14:14:58 out, I'll try that.

14:15:02 Now, with bit cast you can build custom array and created that way. But, you know, standard rate was standardized with c++ 11 but really came from boost 10 years before that.

14:15:13 And it's you know an early example of a way of wrapping horrible see and giving it to c++ facade or API to work with the other algorithms. That's not necessary now in c++ 17 we got data and science free member functions, along with the existing data and

14:15:50 data science. Yeah.

14:15:42 Yeah. So, oh beginning End Of course. So, you know, all of those three functions, interoperate fine with a radius and built in arrays also have a specialization for stud swap.

14:15:55 So, effectively, you know that's a, an old range algorithm that will copy even nested array is properly recursive Lee.

14:16:02 I'm drifting off again. So, standard rate in c++ 20 we got standard to array, which will take an array of value and create a standard array from it. But what wasn't noticed is that in the is Luke here, yeah.

14:16:20 Oh yeah, look see So Luke noticed a couple of months ago that the earlier experimental maker Ray was able to make zero sized standard array.

14:16:27 But to array cannot do that because it's converting built in and ready to a standard rate you can't have a zero size built in array. So, so you know one of the advantages of standard rate is that it's got the zero sys specialization.

14:16:45 And then one of the downsides, is it needs to zero size specialization so every array class you right you have to do as he works on

a specialization. And it's a little tricky to write, it's easy to get wrong.
14:16:56 And there's several of the standard library implementations got it wrong and had issues raised against it.
14:17:01 But anyway, I'm rambling again I guess so.
14:17:04 The history then so it got viewed, but it got raised on the liaison mailing list and was discussed, a month ago was August the sixth.
14:17:14 It holiday season. So, that partly member to dissenting voices are not dissenting there was some concern raised at the highest levels.
14:17:25 Worried of WG 21 didn't make it to the meeting, but sent messages to the reflector with, you know, unspecified concerns that when arrays have been dealt with.
14:17:37 When it's been attempted to improve it raised in the past, it's led to difficulties.
14:17:44 So that's what I want to drill down on you know I'm so, so what came out of that meeting is we need implementations we need compiler implementations. So, the paper has been forwarded to EWG, but it's not worth doing, until we have implementation experience.
14:18:00 And similarly on the sea side wg 14. I've got much stronger gating they require implementations.
14:18:09 So, I happen to have a few weeks, open before contract, that's coming up in mid October, so I volunteered to do the planning and GCC implementations.
14:18:22 And that's what I've started on this, this week.
14:18:26 I don't think we have any other clan guys here it was William Moses and the, the array, the young, the calculus guy is like do to have some sessions with them on the VM side.
14:18:39 Oh yeah, so there's a bunch of Tony tables that are worth looking at. So what does it propose the initialization is fairly simple that you can initialize an array from an array, right hand side.
14:18:52 Then copy. If you have to raise a and b. Now you can just do A equals B, and you get the copy.
14:19:00 Then a little trickier perhaps that's placeholder semantics. So, an auto.
14:19:09 We are playing array will declaration will deduce the element type.
14:19:16 Now that interacts a little oddly with existing language features, and there's a c++ 20 edition. That allows pointers or references to incomplete arrays to be initialized from array is with no deduction going on there.
14:19:30 So that's an template deduction with all with auto attack. So anyway, the placeholder is one possible area of dragons. The other possible area is praised initialization.

14:19:42 You know the code for that and GCC is pretty horrible I've dived into that before. And I'm sure the same is true in clang.

14:19:49 And it's different also between C and c++. So at the moment I'm focusing on the c++ implementation of this

14:20:00 array return from function in fact yes that's the main thing that I kind of got into this foreign, and wanting.

14:20:07 So, that change. I believe will be easy enough to implement, but it then leads to Abi, that this is a new API for both C and c++, and for the wider foreign function interface world.

14:20:21 And that's obviously going to take time, you know, there's so many platforms, especially on the, on the sea side of things, and, you know, who are the API people, they think that meeting some smoky room.

14:20:33 So even find out who they are and how this how this happens, is likely to take some time.

14:20:43 So I guess I'm up to 10 minutes so if if there are any questions we can spend five minutes or so before the next papers.

14:20:51 At the end of this will this make or break a regular type.

14:20:56 Sorry Sagan with this make array regular lets me know it will make it mostly semi regular. So, this. So the problem is with array parameter types or formal parameter types.

14:21:10 So, for the liaison meeting. The authors were keen to discuss it, but the extremely strong feedback was just let's not even discuss it because this proposal will be sunk.

14:21:21 If we talk about passing arrays by value two functions. The syntax is broken and see an array parameter in either a function or a template argument list gets adjusted to the point to type.

14:21:37 And there's nothing you can do about that. Actually, there is on the template argument side and this is something I'm thinking of amending revising the proposal with.

14:21:48 So, in a function you cannot declare an argument with dental type auto, you know, auto works but that gives you a decay copy of the argument that will type of auto with an array argument will deduce array.

14:22:01 But then as a function argument that will be interpreted as an array as the element pointer and, but in a template document list, you can actually use deco type auto that's allowed you pass it in a way it will produce an array.

14:22:17 But then the array can't be initialized.

14:22:20 It will be really nice if you could do that with string literal because it gives you for free.

14:22:25 You know strings as template arguments, which is what people will be finally getting with static string, this is you know you can already do

that in c++ 20 but this allows you to use a built in a way to do that, and potentially, you know, arrays of integers
14:22:40 or floating point numbers, or whatever. So in fact, that only requires a tiny change in the wording for what constitutes a structural type, because that is the requirement on non tight template arguments now.
14:22:56 And TTP is I should say.
14:22:59 And so, Has this been reviewed by any c++ groups.
14:23:05 No.
14:23:06 Okay, except the liaison group officially counts as.
14:23:10 Yeah. Okay, so it was passed over. Because, so December 2019 was a was a tricky period because right after that we got shut down. So, if it's not on anybody's radar is this or no I mean I talked to friends about it because he you know he frequents this
14:23:30 group as well as SG 22.
14:23:33 And he's written a rave related papers in the past, you know after the Vla the array study group debacle. He, he wrote the last paper on killing and helping this and what has he said, is he.
14:23:47 I'm not sure so he before the liaison meeting he sent out a strange message saying why are we talking about this, I don't think he realized it had been on the agenda, but also when I, when I said well it's been you know it hasn't been looked up to 20
14:24:02 months he said, That's usual for c++ proposals small proposals like this go in and unless somebody is pushing it, it's not going to get seen.
14:24:13 So, it hasn't had sufficient inspection from language experts, and I would really like to get that done before the labor that really needs an examination from the design people like so this needs to go through evolution.
14:24:29 working group.
14:24:31 So that's so it's it's do for evolution working group, and the implementation is what is required to make it worth going there.
14:24:39 So hopefully, by early October I'll have something by mid October, enough to move it to EWG while that might be true.
14:24:49 Your, yo, yo, you might also be your, you're also risking the idea that evolution can change this significantly on you, or just rejected outright. And then you might not have anything to you might have wasted some time right so that's true.
14:25:04 And, you know, that was flagged up in liaison as well.
14:25:08 But somebody needs to do this, I'm not, I'm not the best I'm a hacker I can dive in. I've done a couple of PC patches in the past. I'll give it my best go yeah and and the fact that the fact of doing it will

allow you to find, you know some some some
14:25:24 of the answers that this paper is posing and, you know, maybe it should smoke, it should smoke out any issues, and hopefully quell the third because you know there is fear, uncertainty and doubt being spread.
14:25:36 And this will get more certainty.
14:25:40 So, at the end of the day, if this works then you're going to be able to copy a raise to raise directly, without going through some sort of loop. Is that is that right that's right and then it happens to kind of mend a whole bunch of standard algorithms
14:25:58 that are the moment it doesn't make sense to pass a raised because they will fail. So you know ranges is pretty good but as soon as you get to multi dimensional raise it fails, or once you go past the level where you need now value it will fail, I think.
14:26:12 So this just makes it more robust. There's also a c++ 23 proposal to allow braced initialization in more standard algorithms, and that will just work with a radius once, once this is done.
14:26:27 And I think.
14:26:27 Recently, that was a cosmetic change that allows you to do square brackets with, with commerce, to be to be parsed properly as an array.
14:26:39 Sorry, I missed that, this, this.
14:26:43 So this is a paper by.
14:26:46 I'll have to dig it out later it was put in.
14:26:49 My name is Mark Coleman.
14:26:55 No, I'm thinking of another paper this is
14:26:55 I braced initialization in standard algorithm.
14:26:59 Yeah, I know. Yeah.
14:27:02 But yes, I'll check check this afterwards and send out links. Okay, thank you.
14:27:08 Okay, I mean this is this is only the start.
14:27:12 You know, we make it a semi regular, and then there's a bunch of other things that improve things more, like this one, you know that what you're getting into, because if you don't put push this through EWG first, and then do an implementation, while that's
14:27:27 that. That is really helpful it snot smoking out the, the Gremlins you, you know, ew, you might totally reject this or change it. So just be careful.
14:27:40 I mean, I mean, I'm interested in any opinions from the group here as well if anybody thinks either this is crazy or this is great, please drop me a line.
14:27:48 If anybody sees any issues especially. Great.
14:27:54 And yes thanks for that heads up Michael Yeah that's it some

hopefully people will have a bit of a chance to look at I mean arrays is of interest to a lot of people, I would say, especially the high performance computing people and of course by influence

14:28:07 the machine learning people, it would be it would be good if you can pass this to people like Mark Coleman.

14:28:14 Yes, I've talked with him in the past. Yeah, see what he thinks, because they you stay as he is still firmly involved in high performance computing.

14:28:24 Anybody who's doing work in the US National Labs, especially some of the high performance computing labs. So sure, Andrew might be interested who was on the call.

14:28:40 Maybe, Devon Devon Lieber at all gone Nevin Yes, yes, so yes I dropped Andrew line earlier in the year actually along with the other. Some of these guys, they have time they might end and they care, then they might actually take a look.

14:28:50 I think you have to tread a little bit more carefully on that on paper on display but as you said this. This area has lots of Gremlins, and there's a lot of people who have historic antagonism towards any kind of work towards standard on standard array

14:29:03 and still the rest right yes it's almost to do it feels, and I didn't make any friends. So, the last time I was here, I had a rant about p 2128, the multi dimensional array subscript operator.

14:29:15 And we should have a one minute silence because yesterday that was voted to call in do. Yes.

14:29:23 So, you know, I probably didn't make any friends by opposing that but in the end I didn't publish the paper, and it's gone in the world hasn't ended. Okay.

14:29:32 It's good to know. Good to have you bring this to our attention, just to make sure. But, but yeah, this is, this is full of potholes I suspect, in some ways, but if it works, it is it will challenge one of our biggest

14:29:46 areas that we've been having troubles with and that's why we've tried creating replacements for story.

14:29:53 There was an attempt that there was an array subcommittee The reason Yes, his name is so prominent involved in it was because yes he cares about this.

14:30:00 And he was actually chairing this array.

14:30:04 Working Group subgroup.

14:30:06 Okay, which got terminated because they ran into this huge problem with this new design of a library type array which can switch between the heat, and the stack, which was amazingly incredibly difficult to implement.

14:30:23 Yes, I tried to follow the history of that I don't have the reflector access, but I followed the history of that working group and you know went right back to the earlier see groups as well to follow the history of this as far back as I could.

14:30:37 And Yano was deeply involved in this as well, too, he in fact proposed one of the proposed solutions. And I don't know if you can get on his attention on this.

14:30:50 Yes, I've tried, I've opened channels with him and his main feedback is, it's poorly motivated. And this just general concern that when it's kind of in the past it's caused issues.

14:30:59 So you've got a lot of. You don't say you got almost everything covered which is good. I written up more motivation, but to me that's not the most important thing I know why I wanted.

14:31:10 And I'm just ahead of the curve, no problem, that's that's why a lot of us are here, ahead of the curve and that's that's okay.

14:31:16 All right, anything else in the group. Anybody have any questions and things like that on this issue

14:31:25 here hearing none.

14:31:26 Well, thank you.

14:31:29 Well, thank you for for letting me get the news. Yeah.

14:31:32 And we're going to switch over to graph fully now so would Andrew, or Phil would like to go first.

14:31:38 I don't know why we left it at.

14:31:41 I'm going to turn off my shares, and you have you guys go ahead and and and share. Can you guys do.

14:32:30 Okay, so you want to go first. Yeah.

14:32:20 Okay, what do you go ahead. Can you can you share it.

14:32:30 I think so.

14:32:18 host disabled participants screen sharing.

14:32:39 Just a few slides.

14:32:43 talking about so if everyone see my screen. So.

14:32:50 Yep.

14:32:50 And we also, as I mentioned, this was sort of I'm kind of a overview of the first part of the paper that, Phil.

14:33:01 And I've been working on the.

14:33:06 And I should also say its absence represents some of this is just represents my point of view on this and not necessarily feel so if you can, you know, speak up if when there's differences of opinion and, you know, some of what I think we're looking for

14:33:24 is some guidance from the group here as to, you know, what's the best direction to go in regards to certain things and so that we can get

the paper wrapped up and move into LWG.

14:33:41 So just as a kind of reminder some of the goals, behind.

14:33:48 Having a generic library in general or, you know, graph library particular, are that you know following generic programming practice we want to abstract or begin with concrete efficient algorithms to generate generic algorithms that are parameter eyes

14:34:07 and can be combined with different data structures to produce a wide variety of useful software.

14:34:16 So, I think airport principles for an actual c++ standard library is that the

14:34:26 resulting libraries would support and embrace modern c++ idioms.

14:34:34 Rai other, you know, other things in c++ core guidelines for instance, that the existing standard library be embraced for graphs.

14:34:46 You know, I think we need to brace the scale of real world graphs. These days, which might be even on a single laptop, might still be billions millions or billions of vertices and edges, and my own view some stats on this is that graphs.

14:35:05 The important part of standardizing graphs for graph algorithms is the ability to traverse structures implied by relationships among data, not necessarily for storing data.

14:35:21 So some anti desert Radha, kind of come from more than 20 years feedback now from boost graph.

14:35:33 Some of the pushback we've gotten over the years is, you know, boost graph is very very flexible has all kinds of features for every imagined possible use case, but also then tends to be expert friendly on a particular property maps, and visitors introduced

14:35:53 a lot of overhead.

14:35:56 In both in terms of learning and use the library. You want to do simple things.

14:36:01 And on top of all that the library itself in use, particularly with large graphs has very poor efficiency and performance, and because it's so peak in many ways it's hard to diagnose what the causes of the performance loss are.

14:36:22 So, one point of view I guess I want to introduce and throw out for consideration by the group is kind of a following that, the c++ standard library, which began as the standard library has, you know, rich set of algorithms and data structures for one

14:36:48 dimensional containers. And, at the original interface of that based on iterator is has grown, or been further generalized abstracted into ranges.

14:37:01 Then the library also has rich language if you like standardized

mechanisms for defining the type requirements, again, you know, based on iterator is ranges.

14:37:13 So, has a standardized mechanisms for defining these types of requirements that form the interface to generic algorithms and that is now has language support as formerly as concepts and the micro claim is a claim is that the standard library, as it is

14:37:32 with these mechanisms and containers.

14:37:39 Already provides sufficient capability to support generic graph algorithms and data structures.

14:37:45 And in particular, we can define generic graph algorithms as range ranges and use that as requirement on a paragraph types of course you'd want to collect these.

14:37:58 These are the range of ranges is the only requirement. And so we want to collect those into particularly particular set of name requirements or concepts, and then

14:38:12 call those graphs or whatever, if you like.

14:38:17 And in many cases, compositions of standard library containers already meet these requirements for graph algorithms, and that if one wants to use graph data structure or third party structure with any kind of algorithms we define over graphs this way.

14:38:36 They can provide, they could do so, simply by providing this library compliant on the interface. This is how things have worked with the standard library.

14:38:45 So the.

14:38:56 I just jumped right into some of this abstraction process so if we look for instance at graph algorithms. This is a representative for class of algorithms.

14:38:59 When we call it JC list algorithms. There's a few things that the algorithm does over its input arguments so BFS takes a geographic gee I'm starting to protect us.

14:39:15 It first enumerate all the vertices. And then for each vertex sets a value in the color property map to be white, and then it puts the starting vertex of the queue and then pulls vertices off the queue for each for kicks it pulls off the q amp visit slows

14:39:34 vertices and specs whether they've been visited before and if they haven't been visited, but some on the queue says, This actually is abbreviated but taken directly from CSRS text.

14:39:49 And so again, kind of the things need to be able to do a new right the vertices.

14:39:54 Use the vertices to random access into containers or vertex properties, if you like, and then most important sort of the reversal idiom

in many many graph algorithms is that given a starting vertex you, the need to be able to enumerate or traverse the

14:40:14 neighbor vertices.

14:40:18 And so one minimalist approach to realize this just in terms of your existing standard library algorithms will be something like this, where we enumerate the vertices by just running through an integral value from zero to the size of the graph, we store

14:40:41 the colors in STD vector. And so we set each one that we visit the white. We put things in the queue.

14:40:55 The neighbor vertices. So something like this will work with graphs that are defined for instance IPS as an SG vector of STD vector of em, or STD vector, vector, but to pull event to double or what have you and we can call this algorithm on compositions

14:41:17 of

14:41:20 standard library containers.

14:41:24 Another.

14:41:26 So if we, kind of, collect those things into abstract requirements. You know we have that an index a tasty graph.

14:41:36 In this case the vertex identifiers are integral graph that the thing that graph is actually using is what we call the Jason CMT in the abstract algorithm.

14:41:49 That in turn is a random access range, because we can index into it. Random Access fashion. And then if the Jason ZGMU gives us board range that we can iterate through to get get the neighbors.

14:42:05 Another principle that sort of underlying that is kind of what is a graph, really. And so sometimes I like to say there is no graph.

14:42:16 And this kind of comes from recent experiences working with large data sets that have graph like relationships. And so if we think about, you know, a graph that might represent circuit, the quantities of interest are was the vertices, or the note circuit

14:42:35 nodes.

14:42:37 And then the circuit elements between vertices, we can convert that into a graph suitable for doing breakfast search whatever if you like by first turning the data which could be strings or whatever.

14:42:57 The point is that vertices have

14:43:01 that vertices, have some kind of identifier edges, our relationships between things in the vertex set, which might also properties.

14:43:12 We can create with library functionality create an index list and then create a Jason, see List, with those indices, and the ideas that you know general user shouldn't be trying to build this and they don't need to.

14:43:29 And this, the adjacent the list really represents structure, not

actual data bomb if we look, there's kind of one important.

14:43:41 Additional aspect in graph algorithms.

14:43:45 Speaking of data, which we can see and other algorithms like like Dykstra that is that we might have properties associated with edges. In addition to properties associated with vertices.

14:43:56 Typically, and I think this is almost universally true.

14:44:02 When we need a property of a vertex, it's actually, or I'm sorry property of an edge. We need that at the same time as we're accessing the edge. So, basically we can store the property information about an edge with the edge in some way or you can store

14:44:24 an index into the corresponding property, again with that edge so when my child have an algorithm for extra that looks like this, where.

14:44:39 Here we have a vertex property map distances.

14:44:44 and then an edge.

14:44:48 Property map or container let's say your range of that we index into with the edge identifier that sword with the edge.

14:44:59 So the requirements that I guess for this type of graph are that are you know basically the same as we had for the previous type of graph, but the difference is that the value type, now that we get when we are traversing neighbors is twofold.

14:45:22 The first is a vertex identifier the neighbor neighbors were tax ID, and the second is an identifier for the edge, and what that is is context dependent so it could be the actual property, it could be another identifier into the edge.

14:45:40 So, In the case of storing indices with edges.

14:45:44 We have our next address would look like this, and we would store notionally in the library would store.

14:45:52 Again the neighbor vertices, as well as the edge indexes us more.

14:46:04 Maybe more generalized approach would be to actually store the

14:46:13 kind of abstract whether we're actually storing weights on the edges or storing indices into another container on the edges by just passing away function.

14:46:27 So, here again we win iterating through the graph, we have an identifier weights returns the prop weights of that identifier returns, what it is and that subsumes both storing indices or storing actual properties on the edges.

14:46:44 And there's different reasons to want to do one or the other, storing the properties on the edges does give you some efficiency in terms of accessing that cash behavior and so forth.

14:46:58 But, you know, these these might be much larger than an index value would be and so you might actually want to just refer back into the

original Angeles to get the properties.

14:47:12 So one important thing also in in graphs and thinking about them as standard library components is we, I think we need to be able to iterate through them or traverse them with idioms that are used these days in c++ so we saw before.

14:47:35 You know we could iterate through the graph just with indices from zero to the size of the graph. We also might want to use

14:47:48 range based fours, to iterate through the graph or iterate through the neighborhood. The problem with just the plain

14:47:58 range base for is that we don't actually have the source vertex anymore.

14:48:04 We're doing that but we have an adapter could provide an adapter around the graph, so that if you actually need that index you can have that in the range base for.

14:48:20 Similarly, we would want to use standard library.

14:48:26 Algorithms on ranges, in particular, maybe standard for each, and as different ways of using that.

14:48:34 One important use case in using yesterday for eight for each is that we can use.

14:48:41 Execution policies to do this do these things in parallel. And I think it's important for efficient graph algorithms that we be able to support this.

14:48:55 In the BTL.

14:48:57 Some of these are they had had similar concepts so that a graph was basically something that she had two associated types of protection scripture and and it's descriptor.

14:49:19 Then there was a concept that just for the first step that we had in with research for new breeding vertices. So there's a concept for Texas graph. So the vertices function returned to range of our text descriptors, and number two C's return to the size

14:49:26 of that range.

14:49:26 Then there were two types of graphs for

14:49:32 doing visiting neighbors one was Jason vertices, where the Jason vertices returned a range of our text descriptors corresponding the neighbors. And then there was an incidence graph that had four functions out edges about degree.

14:49:47 And then sort of the source function which could take an edge descriptor from the source vertex and target that takes a ninja script or for its target.

14:49:58 So, an example using this kind of interface would look like this.

14:50:04 Again, we don't. this isn't the full Biggio with

14:50:11 properly maps and all that and still just use a random access range for instance, but we can get the number of vertices is number two seasons the graph can get the neighbors.

14:50:23 With the vertices of gigas arrange get it right over for the neighbors.

14:50:29 And then we can get the adjacent neighbors, and iterate through, through those without function,

14:50:42 go quick through. There's a couple of, there's one other content against reversible and that has to do with edge lyst algorithms, where you actually just might want to traverse the edges that are contained in the graph, rather than in just one by one

14:50:56 rather than traversing them in in the order, you get in.

14:51:01 By visiting neighbors, for instance, so you don't need that particular organization.

14:51:08 And there's an example of customers algorithm for instance, how to do that.

14:51:14 And so then in vgl, there was a final concept of an analyst graph that provided the functionality of edges to give you a range of the edges now managers and then source and target to get the source vertex and target vertex from an edge.

14:51:33 Thing is, we've also been looking at his range adapters. So it's, again, in this thinking that graphs are really good ranges of ranges, very powerful part of the ranges library is this notion of adapters.

14:51:49 And when I guess.

14:51:56 And that, in graph, sometimes when people talk about graph, Coco algorithms, they really just mean reversal patterns. So breath research in depth research aren't really algorithms per se, what you want to do with them.

14:52:10 Depends.

14:52:12 And you just want to diverse and breathless source of passionate and do something and so in BTL.

14:52:18 These traverse the patterns were adapted with visitors, which again, we're very heavy weight and confusing mechanism.

14:52:28 So, what we're proposing instead is range adapters that present range of vertices, or edges in order. This particular algorithm to traverse them. So an example for instance of the use of this and this can also shows you know the use of algorithms.

14:52:47 Just over a container of containers.

14:52:49 So you could, you know, given a list of CO stars in different movies.

14:52:57 And, you know, an array of movie stars.

14:53:01 We can nationalize the Jason's team structure go with that.
14:53:06 And then we can just reverse that graph of CO stars in BFS order, and given each edge, and this would be if I said range to get the judges and visited.
14:53:22 We can set the Bacon number for that each vertex has visited. And then we can go through all the actors and report with their bacon numbers are.
14:53:38 So other important utility functions are basically for constructing graphs that before you know there's really no reason to build and Jason structure oneself.
14:53:49 So, given sets of vertices and pairs of vertices or property Associated Press vertices we can easily make those adjustments the structures.
14:54:01 We can also make the transfer of the JC structure, so you might want the, you know, the edges in one direction versus edges in the other direction.
14:54:13 And then, so one particular type.
14:54:31 Then we think it's important to include in this sort of adjacent to structure is a compressed graph, type something got compressed bars row.
14:54:28 It isn't actually a composition of containers like we were looking at before, but can provide that same range of ranges interface, and this is highly efficient, it's in benchmarks we've done it's using this instead of saying vector vectors is about 40% 40%
14:54:46 faster.
14:54:48 And then finally, one thing to, we like to maybe get some feedback on is the importance of by apartheid graphs. And the reason I mentioned this isn't is more, because they kind of come up naturally and even the simplest example of of the IMDb doing Kevin
14:55:11 Bacon numbers, and that really what you're given as entities are movies and actors and the association, that's given for instance by IMDb is just movies and actors have relationships, so your edge list is really between movies and actors, and that really
14:55:30 gives you two different Jason t lists that don't quite meet the requirements we have before. Because there's a different card and out, you know, the things that are being stored in the Jason's the list index into the other or Jason to listen to kind of
14:55:45 cross reference each other.
14:55:49 reference each other. And then that one kind of given that type of decency list. It's easy to provide a library function that would create the actor actor movie movie.
14:56:02 Actually actual graph.

14:56:05 So there's a lot of options and this is I think will be one difficulty in thinking about a graph standard graph libraries what algorithms to actually include, I think it's because there's so many different algorithms and so many variants of them.

14:56:21 I think we need to make it easy to build those.

14:56:25 So, the community can share and build on them, rather than standardizing very many of them.

14:56:34 However, one building block that has emerged, and this might be

14:56:43 a subject of untold talk or. I know, Scott McMillan who's been a proponent of this will be talking about graft laws at CPP con. The idea is basically to generalize there certain linear algebra operations, based on the correspondence pretty graphs and

14:57:00 sparse matrices and kind of the core operations, there are generalized sparse matrix by density vector product and a generalized sparse matrix by sparse matrix product.

14:57:11 Given those one can compose those operations and create sets of algorithms like some of the ones that might, one might see in the water so this is another set of building blocks, one could imagine.

14:57:27 Instead of just the traversing and so forth, we talked about so far, so that I can turn it over to

14:57:36 feel.

14:57:39 I can stop sharing and.

14:57:45 Yeah.

14:57:45 Yeah. Is there a sauce about at this any feedback.

14:57:49 Before I started, was it was it any, any policy wanted to, to catch you out of the slides was at the operations or anything like that that you wanted to highlight.

14:58:01 I'm just trying to figure out

14:58:05 if you were trying to highlight the, the need for certain operations.

14:58:13 Yeah,

14:58:16 maybe.

14:58:27 Maybe it's some sort, some sense ease.

14:58:42 And, you know, some of the design then once the concepts, I guess, are these requirements operations are collected is, you know, what are the

14:58:55 subsequent.

14:59:01 What's the subsequent syntax for realizing these, so one approach and and you know this was kind of this minimalist approach.

14:59:13 We just use range, and you know container syntax to access things that might not be, you know, little sufficient or might not be desirable by the group or the community, but I think what's important are these.

14:59:31 And I think, well, we're bill will jump off from is, you know that we have these certain requirements that the output of what algorithms need.

14:59:56 And, in fact, the the implementation that's there would agree with you.

15:00:02 I think one of the things.

15:00:05 So Andrew and I come at this from different perspectives so he's really been in the algorithms a lot.

15:00:10 I've been on products that have a graph embedded in it.

15:00:14 And so those brings unique challenges from both of those.

15:00:20 So I think both are valid and both are interesting.

15:00:23 I think, in, in this expression of stuff I really like the sickness, and brevity of being able to express the buttons there.

15:00:32 And I've always looked at this and is, and.

15:00:36 And the same.

15:00:38 Can we get there.

15:00:41 To do that, and the other thing that I've looked at the the thing that's going there's there's an emphasis on index looked up.

15:00:50 And when I look at the STL, there's an emphasis on using iterator as the core element here, and. And so this is really great because I think we're getting a lot closer than we used to me.

15:01:07 And I'll show that in a little bit.

15:01:10 But how can we make this most expressive and also.

15:01:15 First of all, so that, for instance, we can adapt existing graphs out in the wild, to the algorithms that are in the standard, for instance, because that's always been a strength of the STL is that somebody that writes this container over here can use

15:01:36 an algorithm that's developed somewhere else.

15:01:40 And so,

15:01:44 there's just kind of my some general observations, I don't think that we're that far off in a lot of ways. Okay, and what what do you think Andrew do you think,

15:02:00 Well I think that's right i mean so they're in one of the, I guess, preface slides I mentioned that

15:02:12 we abstract from concrete efficient algorithms to obtain generic algorithms. And the other part of that is the algorithms that can be combined with different data representations.

15:02:25 So, you know, making sure the you know the the concept definitions are the main type requirements can encompass those different data representations is is kind of important part.

15:02:39 And so, in some sense, it also mentioned. He was kind of coming at this from the point of view of the different data representations. And, and

I was trying to start with, you know, the concrete efficient algorithms.

15:02:53 And also, you know, in some ways I've been, you know, deliberately sticking out maybe an extreme approach in that I wanted to avoid.

15:03:05 You know, getting into the super Swiss Army Knife situation of the, of the moose graph show show you each come from it from a slightly different points of views one is from product ization the other ones from algorithm.

15:03:17 And you are trying to find some unification there. And maybe what we need to do is see what Phyllis is doing now and see if it's getting closer and closer to it to some midpoint, so that it's not particularly leaning to one or the other.

15:03:31 I mean, if it comes to it, we might have, what I'm hearing and I don't know if that's the case that we might have to at some point.

15:03:41 And up have having two different to two proposals, but we wanted to try to avoid that if we can.

15:03:47 That's what linear algebra has has has, has become but, but, but I'm still I'm still hopeful that we can still avoid that, but some sort of unification that.

15:03:55 Yeah, I think so on as Phil mentioned I think we're heading that direction, and he has had his hand up. also I guess. Yeah, go ahead. Yes.

15:04:09 Ok I can hear you. I can't hear you. Yes. Are you are you double muted perhaps I'm double muted, for some reason I don't know why.

15:04:19 So I think the first of all, it seems that, avoiding the booth graph library abstraction layers that have been piled on top of each other there.

15:04:37 Seems like a seems like a good idea. I share the sentiment that exclusively relying on indices might require for example, what you showed in your slides might require that there is some pre processing of the incoming data structure into some index based

15:04:58 arrays or something like that.

15:05:00 And that not everybody might want to do that, just to run a graph algorithm over something that has all the information present, but just in a slightly different, different way.

15:05:15 That said, it seems the abstraction of having a range of ranges. At the core of most of the input requirements is actually a fairly powerful abstraction and that means you should be able to map, most of your data structures that are suitable to run graph

15:05:36 algorithms on in the first place, you should be able to map that to the range of ranges abstraction which might mean to actually, you know, right.

15:05:55 Gosh, view on your data structure. And actually, a to level up

because you need a range of ranges.

15:05:53 Yeah, that's, I mean, if you look at some of the standard library views that come with a lot of Jewish method its writers and sentinels and crap that that might be slightly interesting to do, but at least you know there are literatures fillers looking

15:07:20 Okay.

15:07:35 Okay. Hopefully everyone can see what I have here.

15:07:23 I'm just, it's been a while since we've looked this up but just to remind everyone, I have, you know, played, we're seeing your entire screen, and the actual text you want to show is taking up about one third of the screen.

15:07:36 It would be very helpful to fix that by either resizing your window and and or just sharing the window you want to you want to share with us.

15:07:49 Thank you for

15:07:55 this better.

15:08:05 It's, it's slightly better Yeah, yeah, no, it's getting even better. Okay.

15:08:09 Okay, so, you know, there is an abstraction for, say, Dextre, which is one of the things that Andrew was sharing and so there's the idea of having a function called a distance function its weight function.

15:08:27 It provides the same feature. And so I think,

15:08:32 you know, I, we haven't been looking at that as much here in the recent reviews, because I think it's in large part, Pretty good stab at it.

15:08:44 And I think, in all of the reviews that we've done it.

15:08:50 There hasn't been any disagreement with any of this.

15:08:54 Okay, so just, just as a reminder.

15:08:59 And so I think a lot of the, this has been trying to figure out the right abstractions for representing the data models.

15:09:08 So, And that's, and it's dealing with the concepts and the types. And what does that mean and giving that abstraction right and that's a lot of what I'm going to be talking about today.

15:09:21 And also wanting to do a little bit of a demo of how it would be used.

15:09:27 And I think you'll see how it's, it's coming closer to what Andrew was just shown.

15:09:33 It's not completely there yet, but at least I think we'll be able to see where that's out.

15:09:41 When we were reviewed this last, there was a pretty strong recommendation to reduce the size of the paper. So, at the time it was 70 pages. And so now, and then we're recommendation of making changes so I've

brought those in as best I can.

15:10:02 We're so now we're down to about 46 pages.

15:10:07 Still more than I think people would like, what is definitely improvement from this, I've been doing a lot of work in how types are defined, how the type aliases are defined, how the concepts are defined.

15:10:25 And, and so I can say that the graph traits structure is now gone.

15:10:30 All of the function prototypes are all gone. And the concepts are simpler now.

15:10:37 And so, the.

15:10:43 I've removed those things have removed the mutable concepts for adding and removing things I've removed the inward and outward outward ranges, because we're just not referencing them in any of the algorithms, even though I believe that's important element

15:11:02 to describe graphs in it. I can't justify it. I've also removed that ordered parent I know repair that were there, so that's all contributed.

15:11:13 And so, Let me put that on this on pause.

15:11:20 And I just want to go to the demo.

15:11:31 See if I can find the right.

15:12:13 You're not sharing anything, the smell and I'm trying to find it. I have too many windows open.

15:12:27 Try with less main memory than your program is a new computer crashes if you have too many windows open.

15:12:35 All right, there we go.

15:12:49 Can you see that okay

15:12:49 should be showing a void demo. Yes.

15:12:46 Alright. So, If we start with us.

15:12:52 I'm using Andrews example of a vector for lips.

15:13:00 And in this case, I have a vertex key.

15:13:05 And in this case, it's just the target vertex it's what it is. And in this case the edge is just that target vertex. And I have a vector of a forward list.

15:13:16 Okay.

15:13:18 And then I can define my graph.

15:13:29 The contents aren't that important. Really, we've got four vertices, with five edges. In this case, and then we can if we want to do that, we want to live through that.

15:13:43 We can then create

15:13:46 some

15:13:51 loops.

15:13:52 Now what I've done is I've all of the functions that are in there

are now customization point objects CPS.

15:14:01 And so, in this I've been able to define some very reasonable default.

15:14:09 So there's no special behind the scenes things I'm doing for this demo

15:14:32 Worse, you have a problem with us. You know I went through this.

15:14:50 Well I guess I can't run it, but you get the idea. It ran, just an hour ago, without any problem.

15:14:58 You can see the vertices that I'm getting the vertices, out of the graph.

15:15:03 And in this case, it's just saying, I can take this off right

15:15:12 now. give me the same results.

15:15:16 Even though I can't prove it with pilot.

15:15:20 So in this case it's just saying if it's a range.

15:15:31 I'm.

15:15:50 Suzanne is asking what are the five, five inches.

15:15:55 Oh, I see. It's good because I help.

15:16:05 So the edges are 021 or two little ones over 2122123, and three to zero.

15:16:19 Right

15:16:19 now I can take this back off and I can show you that it's the same exact thing.

15:16:24 So these are customization point objects. So there's a default behavior but then you if you've got your own graph it, you can override this for your own situation.

15:16:37 And then the same thing for the address and these are the same functions that I've been showing for over a year.

15:16:44 And in this case, we're just counting what's there.

15:16:48 Right.

15:16:50 And again with the edges. It's just a little simple rapper.

15:16:58 To do that, and so it can do that by saying, Well, if it's like on the vertices if it's a random accessing range.

15:17:08 Then I'll just give you.

15:17:19 I'll just treat it as a range. As the vertices.

15:17:14 Okay.

15:17:18 Now the question then is, does is it just going to work without any problem, and almost, because in this case.

15:17:29 If I want to get a target key. There will be a problem.

15:17:35 Because I don't know what this is, it's a property on on the edge. And so how do we work around that.

15:17:45 Well, we can define a

15:17:52 function.
15:17:59 We really want to do is just say this is
15:18:08 our duty to get that Ed twitches, are these values here.
15:18:14 Well that doesn't quite work, what we really have to do
15:18:23 is put in the stood namespace.
15:18:28 And the reason that we're having to do that, is that it's using our argument dependent look up to, so it's looking, and that's where the vector lives is in stood namespace.
15:18:40 So this is not a good example, don't do this at home, don't use it in production code that you give us an idea of what how this works.
15:18:49 So anything that works, that
15:18:53 is a property. Within the graph is going to require customization. That's the one thing for sure that you need. There's also a source key, or I'll bring this up.
15:19:12 There are the four functions that really need customization.
15:19:16 If you need them.
15:19:19 If you, if you want to source vertex on edge, you would have to define that and store it.
15:19:26 The other three are really optional.
15:19:30 It really depends on what algorithms you want to use and what how you're storing things on on your graph.
15:19:38 But everything else I think is pretty reasonable definitions and later on when we look at the concepts I've put in there, what those, those default. The default behavior is.
15:19:52 This is just a start, because also within the customization points,
15:20:00 skip down here.
15:20:02 Um, there's different ways of representing that.
15:20:13 If we look at the vertices,
15:20:20 there's three ways of of executing the vertices one is the default which we just looked at.
15:20:29 And if we.
15:20:43 I'm gonna
15:20:50 I'm gonna do it this way if I have my own special graph, I can create a member function called vertices that takes a graph parameter, and it will look for that first.
15:21:03 If it doesn't find that
15:21:07 it will look for
15:21:12 something that looks like this
15:21:15 is a free function that takes a graph as a parameter.
15:21:21 And that actually. Is that right, yeah. Anyway,
15:21:29 and if you can't find it.

15:21:40 It will finally look.

15:21:44 And it will do a default implementation that looks like this it's, there's not actually a function does it as part of the Cpl.

15:21:54 So Tim the namespace stood name, said graph namespace.

15:22:01 So you can see there's a hierarchy here, that allows for graph authors to override the standard functionality so that we now have a uniform way of getting to these things that we're interested in, in the graphs.

15:22:19 One of the things that.

15:22:24 So, with what Andrew presented, you know he was addressing the additional things beyond the incident scraps but then how do you get that the urgency graphs are there Jason seedless arrangements excuse me

15:22:41 say this all the right way there's the Tracy Ranger this over the edge list ranges.

15:22:47 And so there needs to be ways of doing that.

15:22:50 Alright, so DNS has his hand up. Okay, go ahead.

15:22:56 So what you're showing here is what I would expect. The standard way how ranges related customization point objects work, you first try the member function then he tried he.

15:23:19 And then he tried the

15:23:13 function, and then you fall back to some, some plausible plausible other thing, the OD, the difference would be that in Andrews approach, instead of talking about making vertices then similar customization point objects, you would instead.

15:23:39 Define a view on your graph. And that, and you would then possibly customize the begin and end customization point objects on that view if you really must, but usually you don't.

15:23:54 And then that you will just get you the vertices from using whatever technology or or means as necessary to inspect your craft to get those lists, and possibly defining specialized results and all that kind of stuff.

15:24:12 Okay.

15:24:14 I'm hearing what you're saying I'm not as familiar with the fields. So that's a note to look for me to understand that better.

15:24:22 Yeah, so, because what what is returned from this vertices faction. What do you anticipate is being returned from there. From it's a range.

15:24:32 It's a range, right. So,

15:24:37 right. And

15:24:43 why would you scroll up a little bit, you see your edges visited thing, which is very very close to permitting arrangements for loop.

15:24:53 Because the begin customization parent object business is actually what you would get from A.

15:25:02 is more or less what you would get from a range based for loop as well.

15:25:06 Yes, and I'm one of the. So let's look at that.

15:25:25 And we could do.

15:25:40 this reason I was doing.

15:25:43 Copy faces because I've had somebody from there's a lot.

15:25:50 I think right now is not very far from what Andrew has had on the slides right. Exactly. And that's a really good point.

15:26:00 Um, and I've seen that.

15:26:04 But now you see there's a difference because there's a, this is a reference. Right. should be a reference.

15:26:16 And, and so can we go farther to achieve Anders goal.

15:26:24 And that's part of the challenge here. And that's kind of the next step is how far can I go.

15:26:35 You really need the iterator for your further explorations, or is it enough to get to the values, I mean, the shortcut Andrew took for that was, I have editor indeed indices all around them, don't care for the iterations.

15:26:54 So, say that again.

15:26:57 The bottom range based for loop treats in values, your loops above deal in integrators. Correct. The you is an iterator above, it's a value below, and pretty presumably.

15:27:15 Yeah, so, and Andrew gets away with dealing and values because everything is an index, an integer index. Right.

15:27:25 Right. And so at this point now UV is actually an,

15:27:32 an index value. That's the target key.

15:27:36 Right.

15:27:37 And so now, at this point, I'm, I'm having to deal with an index to find the target value.

15:27:46 And so there's kind of this mismatch between dealing with integrators and index says that, that where I'm finding it difficult.

15:27:59 And I haven't put as much time into it.

15:28:02 Okay, so, just from a from a beauty perspective, doing arrange base for loop to run over the vertices is certainly the preferred way of doing it.

15:28:14 Well I think it's also important for, well, to be able to use STD for each, so that it can be done in parallel.

15:28:23 Yes.

15:28:28 Yes.

15:28:30 Yeah.
15:28:32 So, maybe the next step in that area but I don't want to interrupt your presentation for too long. The next step in that area would be to investigate what you can do to make that match was for work, and actually the, and actually was this level of abstraction
15:28:50 and the concept of us, it's not.
15:28:53 I mean, the you in there is not really required to be.
15:29:00 I mean, to be
15:29:05 a,
15:29:08 an index value of sorts or something like that, it could be some more some more complicated data structure of that actually gives you what you want.
15:29:17 Now, Andrew was also doing things like this where he was returning to a tool that had multiple values. Yes, sure. How are you feeling about that.
15:29:34 I don't. I personally.
15:29:38 I don't think that's a problem, I think it's natural for. If you think and abuse, and abuse, fear, it's natural that you will not necessarily have iterate over scalar values but might iterate over more complicated values such as couples, or ranges in
15:29:59 this case.
15:30:01 So that seems seems plausible.
15:30:06 So, yeah, why not.
15:30:13 Okay.
15:30:11 I think that opens up the possibilities for doing that. Okay.
15:30:18 Well, we'll show examples for algorithms that actually need that feature because they don't not only need to iterate over the values but they also need the index of the source, I think, which wasn't otherwise available.
15:30:31 Right.
15:30:33 Right. And I think, essentially do is you.
15:30:39 We actually have a views colon colon enumerate that actually takes a view of our range of values and ads, as an integer index number as a couple, and iterate essentially over a pair of index number plus the value the original view would have given you.
15:31:01 And that's exactly what Andrew did with his key thing, except that we have already made you with that gives that for you from the standard library know invention necessary.
15:31:13 So, what's your thoughts on the center.
15:31:20 Well I mean I, as I had in the slides I think having.
15:31:27 Being able to range based for being able to go quickly, you know,

be able to do things with in parallel, is important.

15:31:37 So, you know, using kind of values or references rather than

15:31:44 generators.

15:31:46 I think is preferable. And

15:31:49 so the slower for lupus of more attractive thing.

15:31:54 Although I would probably still pass in you rather than ampersand you edges.

15:32:02 Yeah, I remember, of course.

15:32:09 Okay.

15:32:12 Okay. Now this really good, I like this.

15:32:19 Okay, that's, that's really all for the demo I want to go back

15:32:25 to the paper.

15:32:33 And I guess what it well fields.

15:32:36 Switching back. One quick comment I didn't list on the requirements on the slides but there are complexity requirements as well on some of these things for you know so that graph algorithms will realize their stated computational complexity.

15:32:53 So, looking up a neighborhood or whatever, for instance, has to be constant time.

15:32:58 Random Access.

15:33:02 Well, I think in terms of that, then that's how it's designed, but if you've gotten something that would work that is less than that, it might be just as attractive to use it.

15:33:17 Rather than having to create a whole new data structure just to do the same thing.

15:33:22 As long as you properly formulate the resulting complexities of the algorithms that you apply to your sub optimal data structures appropriately so that you don't promise complexities that, you know, don't agree.

15:33:42 And of course you can always claim that some kind of look up is sort of constant because usually we would formulate complexities in terms of number of lockups the graph data structure or whatever.

15:33:56 Well, if you look up itself a slow them. Tough luck.

15:34:02 Right.

15:34:03 So now when we switch over here, look at the the ranges, the types and the concepts that are there. So, we've been looking at what we'll see now is, it's gotten a lot shorter.

15:34:25 We have one vertex range, and three edge ranges, and I've thrown in the iterator types as well as convenience, just because I was finding is using them quite a bit.

15:34:40 So, The difference here is now is dependent these types are

dependent on say the vertices function, whatever returns benefit of this is now we don't need graph traits.

15:34:53 And we can rely on the type returned by the function.

15:34:57 We also don't need separate conquest versions of all of these, which was a goal at hand. So this is all working very nicely.

15:35:07 It's very succinct. In terms of the ranges to represent what we've been talking about.

15:35:15 So all of these were there before I've just taken things out.

15:35:20 The other things, the other types that we're talking about are the value types. There's an optional graph vertex and edge value type defined by the user.

15:35:29 There's also a key type vertex key type. And then, just the vertex type itself.

15:35:38 And then there's niche type and the edge key type

15:35:50 is the grace values, sorry to interrupt is the grace value function supposed to do something other than returned.

15:35:58 Other than using the return type or is that actually returning some sort of value.

15:36:04 Well, it's going to return a reference or value.

15:36:10 Sure, but but what graph value of G doing.

15:36:18 I don't know what kind of value would return on the entire graph. So I'm asking. Well, I mean it's just so we have three different types there's a graph engine vertex right.

15:36:28 And sometimes you want user defined values on these objects.

15:36:33 Right. And so, for the context in your context of what you're doing.

15:36:38 And this is from my experience in in working with graphs in a domain area.

15:36:47 You're not going to find a graph value in.

15:36:53 In the algorithms, you might find an edge value.

15:36:59 And.

15:37:01 Yes.

15:37:01 Right. And so it's just returning a value on the edge.

15:37:08 Yes, this one. Sure, but what does graph value do Is it is it supposed to return one value or a tablet maybe that is attached to the entire graph somehow yes to both.

15:37:24 Okay. Okay.

15:37:25 Fine. Seems unrelated to the graph algorithms were talking about that fight.

15:37:30 Okay.

15:37:32 Okay.

15:37:33 Okay. Yeah. Okay.
15:37:38 Any other comments here.
15:37:45 One thing that's because of the way the types of defined, it's no longer quite as explicit.
15:37:57 So when we get in, we have some basic concepts that we have to define and so I've created this vertex iterator. So what is a vertex iterator is something that you can get a key from.
15:38:10 There's a vertex range.
15:38:14 That is a forward range, at minimum, in society size range, and it has vertex iterator so that it has vertex key that you can use.
15:38:26 It has the vertices function, and a fine vertex function, which could just be an array look up.
15:38:36 Again, you can start to see where the default behaviors that I've defined here for this.
15:38:48 There's an edge iterator,
15:38:53 which is a forward iterator.
15:38:57 It has a target, and that target key function.
15:39:02 And so its target key is user override required.
15:39:09 There's a source edge iterator Andrew was pointing out that, and a lot of things you don't need to know what the source key is, are you don't have to store it.
15:39:21 And so, let's pull that out as a separate concept so that's what I've done.
15:39:25 There's the source and the source key,
15:39:29 there's this thing called the other vertex.
15:39:34 that says in when you're an undirected graph.
15:39:41 There's no order to the vertices on an edge, and so you need to know where you've come from that source or source key, And that's what that's for.
15:39:54 When you have. And then there's the edgy.
15:40:01 There's the edge range concept, which has a Ford range, and with an edge iterator.
15:40:12 And then the source to edge range, go with the source edge iterator.
15:40:18 So once we have those now we can start to describe what these graphs are a vertex list graph is a vertex range. I think under has a different a little different idea of what that might be.
15:40:31 That's just focused on the keys.
15:40:34 There's an incidence graph, which is the vertex range. the edge range, and it has an address function.
15:40:46 Now, in a sense, there's some duplication here because the edge

range calls the edges function.

15:40:53 And so, in a sense, I don't really have to define that. But I think it's helpful to be clear on the design.

15:41:02 So, once we see those, the interesting see graph is the same exact thing. The only difference is that range that we're using the explicit range.

15:41:12 And it's matching function.

15:41:15 Same thing with the edge of this graph and the edge range, and the edges.

15:41:25 There's finally an adjacent to the matrix and it's just false. There's some algorithms that can get benefit, knowing that it's working on a matrix. and one of the tears that way.

15:41:37 And so, this is a way to tag your graph, of being initiation two matrix.

15:41:47 There's some contains concepts which are contains vertex contains edge.

15:41:54 It can be useful to have that kind of function available.

15:42:00 Especially when you think of an ACC matrix, it can be a constant time function.

15:42:12 There's the find edge concepts, which is just one per edge type that is just defining a couple functions there's a fine vertex edge.

15:42:28 Based on iterating hours are based on keys.

15:42:34 The same exact thing for the other ranges.

15:42:45 There's the path and cycle concepts, those haven't changed from the past, that's just a less to pass or a list of integrators.

15:43:07 Um, there are some functions that are not in concepts and I've covered most of these there's the graph value of vertex value and adds value.

15:43:06 And the reason I don't have concepts for those is because really the way the algorithms have been designed is that they'll accept the function.

15:43:14 Now they may refer to say an edge value, and that's fine, but they're not the algorithms themselves are not dependent on the presence of the function.

15:43:47 There's another edition which is the degree function is a convenience function that would just return the size of, say, the, the incidence edges or the adjacent see efforts.

15:43:47 It all depends on whatever is appropriate.

15:43:53 Again, these are all customization point objects that can be overriding

15:44:01 over coverage.

15:44:04 And so there's a section on the customization point on FX we've really covered all of the content in here with the demo.

15:44:13 There's another point, and you know we've talked about the agency, less than the edge list.

15:44:22 There's no, it would be very helpful to have adapters that would take the vertex range, and the incident range and create

15:44:36 ranges for these things. The straightforward to do, but I'm trying to keep this simple and keep the pitch count down.

15:44:52 And that's, that's really all the changes, haven't touched the rest of the paper, the algorithms BFS DFS for me says is.

15:45:01 And there we go.

15:45:04 comments.

15:45:14 I guess the only thing I might have a.

15:45:17 It sounded like you wanted, you were trying to make the changes Andrew was looking for but you just wasn't sure how to do it, especially using views. Do you think you might have something that that can work now with Jen's input for youngsters his hands

15:45:32 up next anyway so maybe he's answering that. Go ahead. Yes.

15:45:37 I cannot certainly.

15:45:40 I certainly have no opinion on whether Filton do something with my input or not, or whether my input is helpful.

15:45:49 But, yeah, so I I just feel that the current Phil's current design is closer to booth graph in terms of doing a calling all kinds of mapping functions and in this case customization point objects and all that.

15:46:10 In the middle of it that

15:46:15 that Andrew was explicitly avoiding because it felt to me that he learned from what was crafted, and that didn't actually wasn't very efficient and at some point that it was just some abstraction penalty that couldn't be optimized the way.

15:46:36 So I the the flip side is of course that Andrew gave up on on arbitrary data structures and just said, there's integer indices all around. And that's what you deal with, and there's helper functions to, to actually create those ensure indices lists and

15:46:55 from, from other data structures that you may have. And you keep going from there.

15:47:03 So, I'm.

15:47:07 I'm.

15:47:09 I'm curious to see whether there's a middle ground here, that might abstract away from the integer indices.

15:47:16 On one hand, but does not have the, the performance penalties that apparently bootstrap had or has it at least some use cases, so.

15:47:31 Well,

15:47:35 I'm not sure that I see where there would be a performance penalties in the things that I've done.

15:47:44 But maybe I'm wrong.

15:47:49 So I'm not sure, but I would like to hear what other.

15:47:54 So far, you haven't. You haven't accumulated enough layers of abstraction that would get you close to bootstrap library.

15:48:07 Okay, was Lucas his hand up, and then I want to see what he's what he thinks, how would how you would evaluate it, go ahead.

15:48:15 Thanks, Mike. I just had some comments that really expect any reactions. Yes, but I feel like it's dangerous to do any of this particularly for graphs, in the absence of a really

15:48:28 deep analysis of parallelism parallel applications parallel algorithms on these graphs and how that might work, particularly in sort of where we live in where we're going, either via sort of explicit parallelism in the in the API's or, you know, playing

15:48:43 nice with the existing parallel SPL constructs, which is something I understood a little bit of but I really think that, you know, anything to be universally useful.

15:48:53 Given that the standard has parallelism in it now, we really, it really needs to be, you know, part of are addressed in this particular effort and in addition, like some of the more modern kinds of parallels and things like co routines and the executioner's

15:49:12 work that's going on right now that you know they're not standard yet but it looks very likely that we will have some form of executed parallelism and things like that I think will be very very important and need to be at least plan for or, there needs

15:49:26 to be space in this in what we're standardizing now, so that when those things come online, they're compatible with what has, you know what has gone through us I mean this is a successful effort, I was really, really all I want to say is the I can't forget

15:49:43 the parallelism.

15:49:46 With these very large data graphs and things.

15:49:52 I think with what I've shown today it's really more on, you know, providing a uniform access to this structures.

15:50:03 In terms of the algorithms, I think what you're really addressing is the algorithms. And, and so what I've represented is different ranges, you know, these things edges or vertices or just ranges, as two plus plus or a step defines them.

15:50:22 When it comes to the, the algorithms, it's really or the parallelism is really on the algorithms.

15:50:32 Based on the ranges, and it has to do with just defining an additional parameter that says you know, that gives an option for doing parallelism, is the view that I've had, or I don't disagree i don't disagree at all I guess all I'm saying is things like

15:50:49 are we using iterator is are we using indices, you know, when you start to think about how parallel algorithms are off and implemented those things matter.

15:50:59 And so it was really just an observation that, you know, I feel like if if I can't use it in parallel, I don't want it, that's sort of a personal opinion if I can't.

15:51:09 If I can't use it on a modern, you know, system in parallel going parallel parallel kinds of algorithms, then it's not helpful to, to me, so personally.

15:51:21 Right. Well, not all graph algorithms are are have good parallel answers.

15:51:29 And maybe you should supply them if they can't.

15:51:34 Here's it here's the standard algorithm that can't be done efficiently, right doesn't seem like a good plan in parallel, I mean I understand what you're saying and I'm, I'm a unique user right I only really care about parallelism.

15:51:47 I don't think you're that unique I care a lot about parallelism as well. I mean I don't care at all about sequential algorithms, I guess.

15:51:56 Well let's make it unique.

15:51:59 I'm guessing Luke, you care about efficiency. I do I'm joking.

15:52:03 I'm not trying to be. I know when the transcript comes out, it should be it. You know the nuances got I'm joking, but I am saying that I really do care about the parallel applications of these right well.

15:52:15 The interesting thing about algorithms is that some of the sequential items are faster than the attempts at the parallel.

15:52:21 So again, I think what we all care about is efficiency and graph of it.

15:52:26 Okay, yes it is waiting on line going again.

15:52:30 Okay, yes this is waiting on line, go ahead. Again, I just want to point out that I do agree with the parallelism concerns, in general, we should. However, this particular graph paper should not try to do more than what the usual.

15:52:46 You know one dimensional algorithms and the standard library already support which is a single parameter was an execution policy.

15:52:52 So, and and the algorithms we already have are for example don't work with routines full stop.

15:52:59 And there should be no attempt for the graph algorithms to try to

somehow over with Cortines.

15:53:06 Just don't do it at that level that's a totally different discussion, and we should first discover how the existing standard library one dimensional algorithms work as core routines or can be made to work with authorities before we even tried to integrate

15:53:20 that here. Yes, I totally agree and I really was thinking the execution policies in parallel STL, but I was also thinking that in the future we will have other forms of parallelism as well, that might very well be the case but then I first want to see

15:53:34 those kinds of fellows, apply to the existing standard library algorithms before we burden filled with somehow doing clever thanks for the graph library.

15:53:48 That's all.

15:53:50 Thank you.

15:53:51 Anyone else. I think we should have Andrew have the last word to see if his vision is go ahead and do little things.

15:54:00 Yeah.

15:54:03 So a couple of comments I guess one of you know regarding. Usually, when you presented grasses, you know, based on indices a lot of this did come from, you know, if we really do define a graph, as it's a range of ranges or really a random access range

15:54:23 of forward rages that implies immediately a lot of things that I think lead to using indices, so prisons, cups of time look up and having a different type and all these sorts of things.

15:54:42 So, I think you'll get the kind of the point of view of

15:54:51 coming from is that the data are are separate from separate things from the graph.

15:54:59 And that know we're really just trying to capture and be able to iterate through structure that's implied by certain kinds of relationships among the data.

15:55:13 And I also can echo Luke's point that, you know, being able to do things within the context of the standard library parallelism is really important.

15:55:25 The.

15:55:43 And this is, again, you know, one of the themes I guess are we're sort of principles is or, you know, maybe the thought experiment I was engaging and is can you, you know, does the standard library already provided enough to implement graph algorithms

15:55:47 in a generic way and how, what sort of the answer that yes but then the question is what sort of interfaces, do you want to put on the resulting concepts to meet the class of actual data structures that we want

to use it was algorithms, as large as possible.
15:56:12 But don't you also want to be able to use facilities outside of the standard.
15:56:23 How do you mean.
15:56:25 So if I've got a better vector, or a vector that I like to use, and I'll point to the visit.
15:56:33 What's that one that's being proposed.
15:56:38 Think of cyclone, that's wrong thing.
15:56:42 It just has different characteristics to support stable iterator So for instance, if I wanted to use that or and I've got my own that I'd like to us.
15:56:55 How would I use that. Well, to use me if you're going to use me I guess the question is, is that structure already, is that usable with the standard library.
15:57:06 Now, if it's a provides generators and all of the things that we like. If it's already usable with the standard library, then it should be usable with the graph algorithms that are built using standard library.
15:57:21 So how do we do that. Yeah, and that that's that's the whole point so if, if I have a new thing I guess what a colony or hive structure ago yeah that is usable as a standard library container.
15:57:37 And it meets certain standard library container concept interfaces, then graph algorithms, written to conform to standard library concepts in interfaces, you could drop in any type that meets standard library interfaces.
15:57:53 Right.
15:57:54 And so what I've tried to do share today is to at least with those functions for fine is an attempt to abstract that so that we can use it.
15:58:07 So what other mechanism that I'm sure there are other ways.
15:58:12 How would we do that, um, you know, like I said, the you know, the interface right now the concepts you were defining his range of ranges.
15:58:19 You know colony is a range.
15:58:23 Then, you know, to be able to drop it in, without any difference than, you know what, what's already there.
15:58:32 Okay, so you're going to then. Don't you have to then, also, somehow represent these other types of ranges the adjacent see range and the edge less range.
15:58:53 Are you saying that you would just create views or adopters to, to,
15:58:58 to represent us.
15:59:02 Yes, I mean, again you know the kind of the idea or the theme was, you know, build things using standard library interfaces and since to interoperate with, if you have a new data structure to interoperate with

c++ standard library wanted offices interfaces.

15:59:20 Then, you know, anything, having those interfaces will work with graphs.

15:59:26 Then, I think, you know, there are things though I think as we've talked about, you know, graph data structures that are will just make lists of nodes and the nodes are data, heavy and so forth it.

15:59:47 one of the right ways of providing the necessary interfaces to those. But a lot of that is, you know, I think the interfaces that are defined we shouldn't just give something that's already in c++ standard library, a new name arbitrarily mean sometimes

16:00:01 I think we might want to just to be friendly, you know, because we're talking, are talking about graphs. After all, but I don't think, you know, the lighter weight, the interface can be in the lesson programs have to kind of keep in their brains as they're

16:00:17 using using it.

16:00:20 So what you've, you've expressed things as saying using tools for actors and such, but there's the graphs can be much more wider variety and so you need to, if you want to get that target key for instance, how do you get that out in a uniform way.

16:00:41 Are you going to be forcing things into that structure that you've expressed or how are you going to do that.

16:00:48 Well I mean there's two ways right i mean one is you put views on it, that the would allow access in that way to have the underlying infrastructure, or, you know, you create a more general function to access it.

16:01:07 But then, you know, the user has to adapt to that interface anyway. So the, you know, the question is what interface or the user adapt to.

16:01:19 And if it's easy.

16:01:21 You know if they can provide you know there's some standard interface and, you know, it's already have you were arranged or whatever then putting range adapters on top to provide that view is is one option.

16:01:35 So think about that one more I think we need to have some kind of non composed container examples to work with.

16:01:55 Now,

16:01:58 I'm not sure where to go from my, my focus at this point.

16:02:05 I mean I can continue on, on all of this, but I'm not feeling, getting a sense that there's a clear direction here.

16:02:16 Well, this is certainly a research exercise I mean, in some way, because that what you're doing here in that particular shape and form has not been done before.

16:02:29 So,

16:02:33 the the incremental progress updates we get in these meetings are certainly helpful, but random comments during a one hour video presentation.

16:02:46 Replace, actually, you know,

16:02:51 maybe the comments are totally off because some things just don't work or so, I don't know, but I

16:03:00 I would certainly want would like to see some more.

16:03:06 Some more.

16:03:12 Now, well,

16:03:16 Andrew is one of the things that under does is pre processing the input data into stuff that has indices. And he says, Well, if you have a random access range then obviously you have you have a difference type.

16:03:31 So, you have entity or entities, right there.

16:03:35 And I think bridging that gap is, is important, and understanding

16:03:49 where the problems are, can we bridge that gap by providing a view on some other data structure.

16:03:59 Understand that Phil you don't want a pre processing of your data to just do this index index based stuff that Andrew.

16:04:21 And so it would be interesting to see if there is some way to have a view around your data structures that actually presents to address algorithms interfaces that actually work for example, maybe not, I don't know.

16:04:36 And it could very well be that, and the end of the day, at the end of the day doing the pre processing.

16:04:43 And then running the algorithm is actually faster because, because it's just so much faster to deal with them than more elaborate things than that it's actually worthwhile to do that pre processing step before running the actual process, rather, as opposed

16:05:02 to running the graph algorithm on some complicated internal data structure that just takes it takes comparatively long time to traverse.

16:05:10 You know cache misses and all that.

16:05:16 So,

16:05:16 I don't know, sorry that this is all relatively hazy and not giving you clear marching orders but that's I guess the nature of the problem.

16:05:38 Well I think maybe Andrew and I. One thing that's happened.

16:05:46 Andrew and I've been accepted to do a talk at CPP con.

16:05:52 In a couple months. And so I think that would give us some good opportunities to discuss this more and explore some more.

16:06:09 So I look forward to doing that and it'll be interesting to see how that that comes out.

16:06:21 Yeah, I think that's some.
16:06:23 That's gonna be a good opportunity to see if we can come together. I think the call is done.
16:06:30 It might still be some things you guys can work out.
16:06:32 I'm glad you're called, I'm glad your talk was accepted the, the general talk was was rejected which is fine because I kind of figured that it was a fallback option in case the your thoughts are not accepted.
16:06:44 So, so thank you very much for presenting guys and follows a great feedback and comments, and I will see you guys, unless there's anything else. Anyone wants to see any last things.
16:06:57 If not, Thanks everyone.
16:07:01 Okay.
16:07:02 Yeah.
16:07:08 Yeah, thank you.
16:07:05 And then we'll see you guys in a month and. Cheers. Bye bye.

On Thu, Sep 9, 2021 at 12:41 PM Michael Wong <fraggamuffin_at_[hidden]> wrote:

> *Thanks no worries. I have adjusted the agenda a little bit*
>
> *On Thu, Sep 9, 2021 at 12:11 PM Richard Dosselmann via SG19 <*
> *sg19_at_[hidden]> wrote:*
>
>> *I am unable to attend due to meetings and teaching.*
>> *Quoting Michael Wong via SG19 <sg19_at_[hidden]>:*
>>
>> > *Hi all, SG19 Sept meeting is tomorrow Sept 9.*
>> >
>> > *SG19 Machine Learning 2 hours. This session will focus on Graph*
>> >
>> > *Hi,*
>> >
>> > *Michael Wong is inviting you to a scheduled Zoom meeting.*
>> >
>> > *Topic: SG19 monthly*
>> > *Time: 02:00 PM Eastern Time (US and Canada) 1800 UTC Stats*
>> > *Every month on the Second Thu,*
>> >
>> >
>> > *Join from PC, Mac, Linux, iOS or Android:*
>> >

>> > https://iso.zoom.us/j/93084591725?pwd=K3QxZjJlcnljaE13ZWU5cTILNkx0Zz09
>> > Password: 035530
>> >
>> > Or iPhone one-tap :
>> > US: +13017158592,,93084591725# or +13126266799,,93084591725#
>> > Or Telephone:
>> > Dial(for higher quality, dial a number based on your current
>> location):
>> > US: +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799 or +1
>> > 408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782
>> > or 877 853 5247 (Toll Free)
>> > Meeting ID: 930 8459 1725
>> > Password: 035530
>> > International numbers available: https://iso.zoom.us/u/agewu4X97
>> >
>> > Or Skype for Business (Lync):
>> > https://iso.zoom.us/skype/93084591725
>> >
>> > Agenda:
>> >
>> > 1. Opening and introductions
>> >
>> > The ISO Code of conduct:
>> >
>> https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.pdf
>> >
>> > IEC Code of Conduct:
>> >
>> > https://www.iec.ch/basecamp/iec-code-conduct-technical-work
>> >
>> > ISO patent policy.
>> >
>> >
>> https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Policy.htm
?nodeid=6344764&vernum=-2
>> >
>> > The WG21 Practices and Procedures and Code of Conduct:
>> >
>> https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures
>> >
>> > 1.1 Roll call of participants

>> >

>> > *1.2 Adopt agenda*

>> >

>> > *1.3 Approve minutes from previous meeting, and approve publishing*
>> > *previously approved minutes to ISOCPP.org*

>> >

>> > *1.4 Action items from previous meetings*

>> >

>> > *2. Main issues (125 min)*

>> >

>> > *2.1 General logistics*

>> >

>> > *Meeting plan, focus on one paper per meeting but does not preclude other*
>> > *paper updates:*

>> >

>> > *Sep 9, 2021 02:00 PM ET 1800 UTC Graph*

>> >

>> > *Oct 14, 2021 02:00 PM ET 1800 UTC Differential*
>> *calculus/Reinforcement*
>> > *Learning*

>> >

>> > *Nov 11 , 2021 02:00 PM ET 1800 UTC DST break/cancelled*

>> >

>> > *Dec 9 2021 02:00 PM ET 1800 UTC Stats*

>> >

>> > *ISO meeting status*

>> >

>> > *future C++ Std meetings*

>> >

>> > *2.2 Paper reviews*

>> >

>> > *2.2.1: ML topics*

>> >

>> > *2.2.1.1 Graph Proposal Phil Ratsloff et al*

>> >

>> > *Latest paper:*

>> >

>> > *Here?s a link to the paper (different than the previous paper reviewed).*
>> > *There are some additional updates I?m planning on making before the*
>> *meeting.*

>> >

>> >

>> *https://docs.google.com/document/d/1OpH-xxRri7tJTtJJIZTYmSHkkrZJkdBwm9zJ7LgolfQ/edit?usp=sharing*

>> >

>> >

>>

> *Andrew Lumsdaine' proposal.*

>

>> >

>> >

>> > *P1709R3:*

>> >

>> *https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing*

>> >

>> >

>> *https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing*

>> >

>> > *<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>*

>> >

>> > <

>> >

>> *https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel*

>> >>

>> >

>> > *Array copy semantics:*

>> > *array copy-semantics paper P1997 "Relaxing Restrictions on Arrays",*

>> > *https://wg21.link/p1997*

>> >

>> > *Stats feedback:*

>> >

>> > *P2376R0*

>> > *<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2376r0.pdf>*

>> > *Comments*

>> > *on Simple Statistical Functions (p1708r4): Contracts, Exceptions and*

>> > *Special cases Johan Lundberg*

>> >

>> > *2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva*

>> >

>> > *Reinforcement Learning proposal:*

>> >

>> > *2.2.1.3 Differential Calculus:*

>> >

>> > *https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel*

>> > *2.2.1.4: Stats paper*

>> >

>> > *Current github*

>> >

>> > *https://github.com/cplusplus/papers/issues/475*

>> >

>> > *https://github.com/cplusplus/papers/issues/979*

>> >

>> > *Stats review Richard Dosselman et al*

>> >

>> > *http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p1708r4.pdf*

>> >

>> > *Feedback from Johan Lundberg and Oleksandr Korval*

>> >

>> > *https://isocpp.org/files/papers/D2376R0.pdf*

>> >

>> > *P1708R3: Math proposal for Machine Learning: 3rd review*

>> >

>> > *PXXXX: combinatorics: 1st Review*

>> >

>> >> *std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2*
>> >> *above is the stats paper that was reviewed in Prague*
>> >> *http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19*

>> >>

>> >> *Review Jolanta Polish feedback.*
>> >> *http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html*

>> >

>> > *2.2.3 any other proposal for reviews?*

>> >

>> > *2.3 Other Papers and proposals*

>> >

>> > *P1416R1: SG19 - Linear Algebra for Data Science and Machine Learning*

>> >

>> *https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk7xKikK4Yp8/edit#heading=h.tj9hitg7dbtr*

>> >

>> > *P1415: Machine Learning Layered list*

>> >

>> *https://docs.google.com/document/d/1elNFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr*

>> >

>> > *2.2.2 SG14 Linear Algebra progress:*

>> > *Different layers of proposal*

>> >

>> *https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXIK_d0ljdUAtSQ/edit*

>> >

>> > *2.5 Future F2F meetings:*

>> >

>> > *2.6 future C++ Standard meetings:*

>> > *https://isocpp.org/std/meetings-and-participation/upcoming-meetings*

>> >

>> > *None*

>> >

>> > *3. Any other business*

>> >

>> > *New reflector*

>> >

>> > *http://lists.isocpp.org/mailman/listinfo.cgi/sg19*

>> >

>> > *Old Reflector*

>> > *https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19*

>> > *<https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/sg14>*

>> >

>> > *Code and proposal Staging area*

>> >

>> > *4. Review*

>> >

>> > *4.1 Review and approve resolutions and issues [e.g., changes to SG's*
>> > *working draft]*

>> >

>> > *4.2 Review action items (5 min)*

>> >

>> > *5. Closing process*

>> >

>> > *5.1 Establish next agenda*

>> >

>> > *Oct 14*

>> >

>> > *5.2 Future meeting*

>> >

>> >

>> >

>> > *Oct 14, 2021 02:00 PM ET 1800 UTC Differential*

>> *calculus/Reinforcement*

>> > *Learning*

>> >

>> > *Nov 11 , 2021 02:00 PM ET 1800 UTC DST break/cancelled*

>> >

>> > *Dec 9 2021 02:00 PM ET 1800 UTC Stats*

Minutes for 2021/10/14 SG19 Conference Call

On Wed, Oct 13, 2021 at 1:05 PM Michael Wong <fraggamuffin_at_[hidden]> wrote=
:

> *Hi all, SG19 Oct meeting is tomorrow.*
>
> *SG19 Machine Learning 2 hours. This session will focus on Differential*
> *calculus/Reinforcement Learning*
>
> *Hi,*
>
> *Michael Wong is inviting you to a scheduled Zoom meeting.*
>
> *Topic: SG19 monthly*
> *Time: 02:00 PM Eastern Time (US and Canada) 1800 UTC Stats*
> *Every month on the Second Thu,*
>
>
> *Join from PC, Mac, Linux, iOS or Android:*
>
> *https://iso.zoom.us/j/93084591725?pwd=3DK3QxZjJlcnljaE13ZWU5cTlLNkx0Zz09*
> *Password: 035530*
>
> *Or iPhone one-tap :*
> *US: +13017158592,,93084591725# or +13126266799,,93084591725#*
> *Or Telephone:*
> *Dial(for higher quality, dial a number based on your current location=*
):
> *US: +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799 or +1*
> *408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782*
> *or 877 853 5247 (Toll Free)*
> *Meeting ID: 930 8459 1725*
> *Password: 035530*
> *International numbers available: https://iso.zoom.us/u/agewu4X97*
>
> *Or Skype for Business (Lync):*

> *https://iso.zoom.us/skype/93084591725*
>
> *Agenda:*
>
> *1. Opening and introductions*
>
> *The ISO Code of conduct:*
> *https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.pdf*
>
> *IEC Code of Conduct:*
>
> *https://www.iec.ch/basecamp/iec-code-conduct-technical-work*
>
> *ISO patent policy.*
>
>
> *https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Po*=
licy.htm?nodeid=3D6344764&vernum=3D-2
>
> *The WG21 Practices and Procedures and Code of Conduct:*
>
> *https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedu*=
res
>
> *1.1 Roll call of participants*
>
> *1.2 Adopt agenda*
>
> *1.3 Approve minutes from previous meeting, and approve publishing*
> *previously approved minutes to ISOCPP.org*
>
> *1.4 Action items from previous meetings*
>
> *2. Main issues (125 min)*
>
> *2.1 General logistics*
>
> *Meeting plan, focus on one paper per meeting but does not preclude other*

> *paper updates:*
>
> *Sep 9, 2021 02:00 PM ET 1800 UTC Graph Done*
>
> *Oct 14, 2021 02:00 PM ET 1800 UTC Differential calculus/Reinforcement*
> *Learning*
>
> *Nov 11 , 2021 02:00 PM ET 1800 UTC DST break/cancelled*
>
> *Dec 9 2021 02:00 PM ET 1800 UTC Stats*
>
> *ISO meeting status*
>
> *future C++ Std meetings*
>

C++23 cutoff is now using electronic polling EOY instead of a Feb 2022 F2F

> *2.2 Paper reviews*
>
> *2.2.1: ML topics*
>
> *2.2.1.1 Graph Proposal Phil Ratsloff et al*
>
> *Latest paper:*
>
> *Here=E2=80=99s a link to the paper (different than the previous paper rev=*
 iewed).
> *There are some additional updates I=E2=80=99m planning on making before t=*
he meeting.
>
>
> *https://docs.google.com/document/d/1OpH-xxRri7tJTtJJIZTYmSHkkrZJkdBwm9zJ7=*
LqolfQ/edit?usp=3Dsharing
>
>
>
>

> *P1709R3:*
>
> *https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyY*=
dRy4dM/edit?usp=3Dsharing
>
>
> *https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1U*=
g_vkkE/edit?usp=3Dsharing
>
> *<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>*
>
> *<*
>
> *https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E*=
8PoGS0/edit?ts=3D5fff27cd#heading=3Dh.9ogkehmdmtel
> >
>
> *Array copy semantics:*
> *array copy-semantics paper P1997 "Relaxing Restrictions on Arrays",*
> *https://wg21.link/p1997*
>
> *Stats feedback:*
>
> *P2376R0*
> *<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2376r0.pdf>*
> *Comments*
> *on Simple Statistical Functions (p1708r4): Contracts, Exceptions and*
> *Special cases Johan Lundberg*
>
> *2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva*
>
> *Reinforcement Learning proposal:*
>
> *2.2.1.3 Differential Calculus:*
>

>
> *https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E*=
8PoGS0/edit?ts=3D5fff27cd#heading=3Dh.9ogkehmdmtel
>
> *2.2.1.4: Stats paper*
>
> *Current github*
>
> *https://github.com/cplusplus/papers/issues/475*
>
> *https://github.com/cplusplus/papers/issues/979*
>
> *Stats review Richard Dosselman et al*
>
> *http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p1708r4.pdf*
>
> *Feedback from Johan Lundberg and Oleksandr Korval*
>
> *https://isocpp.org/files/papers/D2376R0.pdf*
>
> *P1708R3: Math proposal for Machine Learning: 3rd review*
>
> *PXXXX: combinatorics: 1st Review*
>
> > *std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2*
> > *above is the stats paper that was reviewed in Prague*
> > *http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19*
> >
> > *Review Jolanta Polish feedback.*
> > *http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html*
>
> *Wg14 Floating point study Group*
IEC 60559 interchange extended types
16 bit floating point type
WG14 N2405
new floating point types to be added to C
exactly the type specified by IEEE, not hex floating point, but binary

floating point, and DFP is also optional

_FloatN, _DecimalN with functions on these floating point types

these are reserved for standard

near done for C23

C++ has P1467 adding similar types for ML

some divergence: new types std::, conversion preference is different:

double+float64 =3D double instead of float64 in C, which is

unobservable except in vararg

might make C++23

> 2.2.3 any other proposal for reviews?
>
> 2.3 Other Papers and proposals
>
> P1416R1: SG19 - Linear Algebra for Data Science and Machine Learning
>
> https://docs.google.com/document/d/1IKUNiUhBgRURW-
UkspK7fAAyIhfXuMxjk7xKi=
kK4Yp8/edit#heading=3Dh.tj9hitg7dbtr
>
> P1415: Machine Learning Layered list
>
> https://docs.google.com/document/d/1elNFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5t=
LVSj64/edit#heading=3Dh.tj9hitg7dbtr
>
> 2.2.2 SG14 Linear Algebra progress:
> Different layers of proposal
>
> https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_
d0lj=
dUAtSQ/edit
>

> *2.5 Future F2F meetings:*

>

> *2.6 future C++ Standard meetings:*

> *https://isocpp.org/std/meetings-and-participation/upcoming-meetings*

>

> *None*

>

> *3. Any other business*

>

> *New reflector*

>

> *http://lists.isocpp.org/mailman/listinfo.cgi/sg19*

>

> *Old Reflector*

> *https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19*

> *<https://groups.google.com/a/isocpp.org/forum/?fromgroups=3D#!forum/sg14>*

>

> *Code and proposal Staging area*

>

> *4. Review*

>

> *4.1 Review and approve resolutions and issues [e.g., changes to SG's*

> *working draft]*

>

> *4.2 Review action items (5 min)*

>

> *5. Closing process*

>

> *5.1 Establish next agenda*

>

> *Dec 9*

>

> *5.2 Future meeting*

>

>

>

> *Oct 14, 2021 02:00 PM ET 1800 UTC Differential calculus/Reinforcement*

> *Learning*

>
> *Nov 11 , 2021 02:00 PM ET 1800 UTC DST break/cancelled*
>
> *Dec 9 2021 02:00 PM ET 1800 UTC Stats*

Minutes for 2022/01/13 SG19 Conference Call

Michael Wong
Luke
D'Alessandro
Scott McMillan
Will Wray
Kevin Deweese
Phil Ratzloff
Ozan Irsoy
Marco Foco
Guy Davidson

> *From:* SG19 <sg19-bounces_at_[hidden]> *On Behalf Of *Michael Wong
> via SG19
> *Sent:* Tuesday, January 11, 2022 10:49 PM
> *To:* sg19_at_[hidden]
> *Cc:* Michael Wong <fraggamuffin_at_[hidden]>
> *Subject:* [SG19] Jan 13 SG19 Monthly Zoom call
>
>
>
> *EXTERNAL*
>
> Hi all, SG19 Oct meeting is Jan 13.
> SG19 Machine Learning 2 hours. This session will focus on Graph
>
> Hi,
>
> Michael Wong is inviting you to a scheduled Zoom meeting.
>
> Topic: SG19 monthly
> Time: 02:00 PM Eastern Time (US and Canada) 1900 UTC Stats
> Every month on the Second Thu,
>
>
> Join from PC, Mac, Linux, iOS or Android:
>
> https://iso.zoom.us/j/93084591725?pwd=K3QxZjJlcnIjaE13ZWU5cTlLNkx0Zz09
>

<[https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fiso.zoom.us%2Fj%2F93084591725%3Fpwd%3DDK3QxZjJlcnljaE13ZWU5cTlLNkx0Zz09&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=SVmhRm%2FLJnMJlazZ7XwlqWhr94SddVSfoMJxA0XrHFw%3D&reserved=0>](https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fiso.zoom.us%2Fj%2F93084591725%3Fpwd%3DDK3QxZjJlcnljaE13ZWU5cTlLNkx0Zz09&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=SVmhRm%2FLJnMJlazZ7XwlqWhr94SddVSfoMJxA0XrHFw%3D&reserved=0)

> Password: 035530
>
> Or iPhone one-tap :
> US: +13017158592,,93084591725# or +13126266799,,93084591725#
> Or Telephone:
> Dial(for higher quality, dial a number based on your current location):
> US: +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799 or +1
> 408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782
> or 877 853 5247 (Toll Free)
> Meeting ID: 930 8459 1725
> Password: 035530
> International numbers available: https://iso.zoom.us/u/agewu4X97
>
<[https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fiso.zoom.us%2Fu%2Fagewu4X97&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=u3tmg3FgukTUc5DQ3zAikZbRYFUkeJTbHkMkj9Kmglw%3D&reserved=0>](https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fiso.zoom.us%2Fu%2Fagewu4X97&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=u3tmg3FgukTUc5DQ3zAikZbRYFUkeJTbHkMkj9Kmglw%3D&reserved=0)
>
> Or Skype for Business (Lync):
> https://iso.zoom.us/skype/93084591725
>
<[https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fiso.zoom.us%2Fskype%2F93084591725&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=CK36IdJux3xcFKIDYVL4iz96tv1jHyZc6boHcR3s7vU%3D&reserved=0>](https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fiso.zoom.us%2Fskype%2F93084591725&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=CK36IdJux3xcFKIDYVL4iz96tv1jHyZc6boHcR3s7vU%3D&reserved=0)
>
> Agenda:

>
> 1. Opening and introductions
>
> The ISO Code of conduct:
> https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.pdf
>
<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.iso.org%2Ffiles%2Flive%2Fsites%2Fisoorg%2Ffiles%2Fstore%2Fen%2FPUB100397.pdf&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=Sdl4Qt%2BeW%2FT0iUj6d6kKMik4qgDoQcyviQ5CgWtFE8Y%3D&reserved=0>
>
> IEC Code of Conduct:
>
> https://www.iec.ch/basecamp/iec-code-conduct-technical-work
>
<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.iec.ch%2Fbasecamp%2Fiec-code-conduct-technical-work&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=NS7744ZJ8LbpHcel7E7s087AK5wFK%2Fz7hUi9ungfUig%3D&reserved=0>
>
> ISO patent policy.
>
>
> https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Policy.htm?nodeid=6344764&vernum=-2
>
<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fisotc.iso.org%2Flivelink%2Flivelink%2Ffetch%2F2000%2F2122%2F3770791%2FCommon_Policy.htm%3Fnodeid%3D6344764%26vernum%3D-2&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIi

LCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=Ay1cmXeGtsBQBz4LPqS3c
TBathDG0u%2FiS15VKGAM8lA%3D&reserved=0>

>

> The WG21 Practices and Procedures and Code of Conduct:

>

> https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures

>

<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fisocpp.org%
2Fstd%2Fstanding-documents%2Fsd-4-wg21-practices-and-
procedures&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af6
1d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C63777
5562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIj
oiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=Nf7uhQaAJPDIm
zkyuWh37WQFVSmHH7CUC3gbBti%2FADo%3D&reserved=0>

>

> 1.1 Roll call of participants

>

> 1.2 Adopt agenda

NO papers are ready today, so short call. No discussion.

>

> 1.3 Approve minutes from previous meeting, and approve publishing
> previously approved minutes to ISOCPP.org

>

> 1.4 Action items from previous meetings

>

> 2. Main issues (125 min)

>

> 2.1 General logistics

>

> Meeting plan, focus on one paper per meeting but does not preclude other
> paper updates:

>

> Jan 13, 2022 02:00 PM ET/1900 UTC: Graph
> Feb 10, 2022 02:00 PM ET/1900 UTC: Reinforcement Learning and
> Differential Calculus
> Mar 10, 2022 02:00 PM ET/1900 UTC: Stats

>
>
> ISO meeting status
>
> future C++ Std meetings
>
> 2.2 Paper reviews
>
> 2.2.1: ML topics
>
> 2.2.1.1 Graph Proposal Phil Ratsloff et al
>
> Latest paper:
>
> Here's a link to the paper (different than the previous paper reviewed).
> There are some additional updates I'm planning on making before the
> meeting.
>
>
> https://docs.google.com/document/d/1OpH-xxRri7tJTtJJIZTYmSHkkrZJkdBwm9zJ7LqolfQ/edit?usp=sharing
>
<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1OpH-xxRri7tJTtJJIZTYmSHkkrZJkdBwm9zJ7LqolfQ%2Fedit%3Fusp%3Dsharing&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=1RfMZVXY%2BfxVdZM3gui3qgg36CXg22S5IZVesNv%2FHtY%3D&reserved=0>
>
>
>
>
> P1709R3:
>
> https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing

>

<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM%2Fedit%3Fusp%3Dsharing&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=HbAK4FnEksTUsclK05OhNcoe2pyBVEmjl3L9JojqNuQ%3D&reserved=0>

>

>

> https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing

>

<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE%2Fedit%3Fusp%3Dsharing&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=35ZizgpoXyWE5PcfxsyHpEjjsJ4zmb5V9nfhoGJyncQ%3D&reserved=0>

>

> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html

> <https://nam02.safelinks.protection.outlook.com/?url=http%3A%2F%2Fwww.open-std.org%2Fjtc1%2Fsc22%2Fwg21%2Fdocs%2Fpapers%2F2020%2Fp2119r0.html&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659841038%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=%2Bui3GMhzb%2BucxCOE2hyPWoGdYqifaE%2F1S%2BF0t%2FaWzck%3D&reserved=0>

> >

>

> <

>

> https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel

>

<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-

_E8PoGS0%2Fedit%3Fts%3D5fff27cd%23heading%3Dh.9ogkehmdmtel&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWWwiLCJXVCI6Mn0%3D%7C1000&sdata=DFACU2uYPRJSpUDv2%2F47mHelClfxbUBXJo%2FaoulR0fI%3D&reserved=0>

> *>*
>
> Array copy semantics:
> array copy-semantics paper P1997 "Relaxing Restrictions on Arrays",
> https://wg21.link/p1997
>
<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwg21.link%2Fp1997&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C6377755562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWWwiLCJXVCI6Mn0%3D%7C1000&sdata=NkZAtdgMWNZj7cZHolSLfXZjHYdxyYiZnmlDdGx319o%3D&reserved=0>
>
> Stats feedback:
>
> P2376R0
> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2376r0.pdf
> <https://nam02.safelinks.protection.outlook.com/?url=http%3A%2F%2Fwww.open-std.org%2Fjtc1%2Fsc22%2Fwg21%2Fdocs%2Fpapers%2F2021%2Fp2376r0.pdf&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWWwiLCJXVCI6Mn0%3D%7C1000&sdata=hSPyvn8beJCTXCxMtBTbI2RnSfHg2YUBFDGlNx2VwlQ%3D&reserved=0>
> >
> Comments
> on Simple Statistical Functions (p1708r4): Contracts, Exceptions and
> Special cases Johan Lundberg
>
> 2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva
>
> Reinforcement Learning proposal:

>
> 2.2.1.3 Differential Calculus:
>
>
> https://docs.google.com/document/d/175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E8PoGS0/edit?ts=5fff27cd#heading=h.9ogkehmdmtel
>
<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F175wIm8o4BNGti0WLq8U6uZORegKVjmnpfc-_E8PoGS0%2Fedit%3Fts%3D5fff27cd%23heading%3Dh.9ogkehmdmtel&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=DFACU2uYPRJSpUDv2%2F47mHelClfxbUBXJo%2FaouIR0fI%3D&reserved=0>
>
> 2.2.1.4
>
<https://nam02.safelinks.protection.outlook.com/?url=http%3A%2F%2F2.2.1.4%2F&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=zw76D0g7VMmxpV0jgvIZU3BUb6DUccoLU2CteoKAwok%3D&reserved=0>:
> Stats paper
>
> Current github
>
> https://github.com/cplusplus/papers/issues/475
>
<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fcplusplus%2Fpapers%2Fissues%2F475&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=%2BgOBfhgFpg1PY0%2Bweo4FQ4HTEisfbZx0OnrvbsEBO6o%3D&reserved=0>
>

> https://github.com/cplusplus/papers/issues/979
>
<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fcplusplus%2Fpapers%2Fissues%2F979&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a430955523373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=5p3jmupSMvTKhUwppQCqWgQUOoj3qJaJ1V9ZRBB3LFM%3D&reserved=0>
>
> Stats review Richard Dosselman et al
>
> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p1708r4.pdf
> <https://nam02.safelinks.protection.outlook.com/?url=http%3A%2F%2Fwww.open-std.org%2Fjtc1%2Fsc22%2Fwg21%2Fdocs%2Fpapers%2F2021%2Fp1708r4.pdf&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a430955523373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=%2B5xlVnQTVTuTkE28iGIxGmk5JZHAWWAd1mlzN3EKWNY%3D&reserved=0>
>
> Feedback from Johan Lundberg and Oleksandr Korval
>
> https://isocpp.org/files/papers/D2376R0.pdf
>
<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fisocpp.org%2Ffiles%2Fpapers%2FD2376R0.pdf&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a430955523373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=yF0O5GtGF%2BypEcbEO0hfuT6yWtLNw22NUUEEGQwtWps%3D&reserved=0>
>
> P1708R3: Math proposal for Machine Learning: 3rd review
>
> PXXXX: combinatorics: 1st Review
>
> *> std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2

>

<https://nam02.safelinks.protection.outlook.com/?url=http%3A%2F%2Fstd.org%2Fjtc1%2Fsc22%2Fwg21%2Fdocs%2Fpapers%2F2020%2Fp1708r2&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=pilyqbqPauLWFWdAwfxX7aTDgdA2zIp9wMjuss4h26Q%3D&reserved=0>*

> *> above is the stats paper that was reviewed in Prague*

> *> http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19

>

<https://nam02.safelinks.protection.outlook.com/?url=http%3A%2F%2Fwiki.edg.com%2Fbin%2Fview%2FWg21prague%2FP1708R2SG19&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=EN20G1AUILxATlpzxk0FnrGJG01m36A%2F%2BLG4SLDp%2Fr8%3D&reserved=0>*

> *>*

> *> Review Jolanta Polish feedback.*

> *> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html

> <https://nam02.safelinks.protection.outlook.com/?url=http%3A%2F%2Fwww.open-std.org%2Fjtc1%2Fsc22%2Fwg21%2Fdocs%2Fpapers%2F2020%2Fp2119r0.html&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=q1NUiW7rvv8t3RQ2swFwDferkRK3ftBiFgUh38D4sT4%3D&reserved=0>*

>

> 2.2.3 any other proposal for reviews?

>

> 2.3 Other Papers and proposals

>

> P1416R1: SG19 - Linear Algebra for Data Science and Machine Learning

>

> https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk7xKikK4Yp8/edit#heading=h.tj9hitg7dbtr

>

<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1IKUNiUhBgRURW-UkspK7fAAylhfXuMxjk7xKikK4Yp8%2Fedit%23heading%3Dh.tj9hitg7dbtr&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=ZKbKkyoTtalkLc6wJwR7T3HmEV7fY%2BfPTiL82%2Ff7%2Bdk%3D&reserved=0>
>
> P1415: Machine Learning Layered list
>
> https://docs.google.com/document/d/1elNFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr
>
<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1elNFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5tLVSj64%2Fedit%23heading%3Dh.tj9hitg7dbtr&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=aHg1E3qfyrDZmxTHERPzjyFrU3431JU8hPNTAOdoh9o%3D&reserved=0>
>
> 2.2.2 SG14 Linear Algebra progress:
> Different layers of proposal
>
> https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0ljdUAtSQ/edit
>
<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0ljdUAtSQ%2Fedit&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=nFjf1tX2AWiE5m%2FA15l7dUQIpppTJJRtrnmyUjpUne8%3D&reserved=0>
>
> 2.5 Future F2F meetings:
>

> 2.6 future C++ Standard meetings:

> https://isocpp.org/std/meetings-and-participation/upcoming-meetings

>

<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fisocpp.org%2Fstd%2Fmeetings-and-participation%2Fupcoming-meetings&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6lk1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=nXwCOGd2pFhoE9BJ0TRsbHsIRfbgFk4ICTXOTX4dEF0%3D&reserved=0>

>

> None

>

> 3. Any other business

>

> New reflector

>

> http://lists.isocpp.org/mailman/listinfo.cgi/sg19

>

<https://nam02.safelinks.protection.outlook.com/?url=http%3A%2F%2Flists.isocpp.org%2Fmailman%2Flistinfo.cgi%2Fsg19&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=k6KX4ARlxffSk5VLIwPiD49lUNGAAnYBczHrD5nCBXQ%3D&reserved=0>

>

> Old Reflector

> https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19

>

<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgroups.google.com%2Fa%2Fisocpp.org%2Fforum%2F%23!newtopic%2Fsg19&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=yivgUYSHwCaIbj%2F%2Fj3wF90F71iV8i23hel2y7tmkf%2BM%3D&reserved=0>

> <https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/sg14

>

<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgroups.google.com%2Fa%2Fisocpp.org%2Fforum%2F%3Ffromgroups%3D%23!forum%2Fsg14&data=04%7C01%7CPhil.Ratzloff%40sas.com%7C166e4c4a8b564e3af61d08d9d57e816f%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637775562659997294%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=0rkgyQ8YvtTLqv0l6QH%2B50%2FAo2k32kKGCN5tKa1FgjE%3D&reserved=0>

> >

>

> Code and proposal Staging area

>

> 4. Review

>

> 4.1 Review and approve resolutions and issues [e.g., changes to SG's
> working draft]

>

> 4.2 Review action items (5 min)

>

> 5. Closing process

>

> 5.1 Establish next agenda

>

>

> 5.2 Future meeting

>

>

>

> Jan 13, 2022 02:00 PM ET/1900 UTC: Graph
> Feb 10, 2022 02:00 PM ET/1900 UTC: Reinforcement Learning
> and Differential Calculus
> Mar 10, 2022 02:00 PM ET/1900 UTC: Stats