Subject: Generating N random numbers that add to a TOTAL
Posted by cgguido on Thu, 07 Aug 2014 03:52:47 GMT
View Forum Message <> Reply to Message

Hi all,

I am wondering if anybody has suggestions on how to improve the function below. It seems ok for floating precision numbers.

For integers it's a different story:
It works great if N<<TOTAL. When N approaches TOTAL I get a few numbers and then a bunch of zeros... Also, setting /DIFFERENT makes it run for ever if N is large. Also, the sum of res adds up TOTAL=/-1, not always to TOTAL exactly...

Suggestions?

Thanks,
Gianguido

```
FUNCTION nrndaddto,  n,  total,  integers = integers,  different = different

compile_opt idl2

res = dblarr(n)
res[0] = randomu(seed, 1,  /double)*(total)

FOR i = 1,  n-2 DO BEGIN
  res[i] = randomu(seed, 1,  /double)*(total-total(res[0:i-1],  /double))
ENDFOR
res[n-1] = total-total(res[0:n-2],  /double)

IF ~keyword_set(integers) THEN integers = 0

IF keyword_set(integers) THEN res = round(res)
IF keyword_set(different) THEN BEGIN
  IF n_elements(res) NE n_elements(unique(res,  /sort)) THEN res = $
  nrndaddto(n, total, integers = integers, different = 1)
ENDIF


RETURN,  res
END
```

Subject: Re: Generating N random numbers that add to a TOTAL

Posted by cgguido on Thu, 07 Aug 2014 04:24:25 GMT
View Forum Message <> Reply to Message

Looking at the output of the above, it didn't seem like the numbers were uniformally distributed, so to speak...

Just had a thought:

If I generate N-1 numbers between 0 and TOTAL, then I could use the intervals between the sorted numbers no?

Something like:

```
n = 4
total = 100
r=randomu(seed,n-1)*total
sr=[r[sort(r)],  total]
res= sr-shift(sr,1)
res[0] = sr[0]
print, res, total(res)
```

Still have problems if I round the result. Don't always get total(res)=TOTAL...

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Russell Ryan on Thu, 07 Aug 2014 12:58:46 GMT
View Forum Message <> Reply to Message

So, the problem with the integers is that you recast floats as integers post facto. If you want integers and you want them to sum to a total, then you need to draw integers up front --- otherwise you're noe ensuring that the round(res) is always summing to total. But this is where you're going to run into trouble...

At each step you're drawing a random number between 0 and the requested total *MINUS* the running total. If you plot the random number as a function of iteration variable, you'll see that the typical value is going down. In fact, near the end of your run, the value will be very small --- because you're converging to the requested total. Therefore, that random variable will often be zero (or 1 and rarely higher). That doesn't seem like a good thing, but maybe it is...

Russell

On Wednesday, August 6, 2014 11:52:47 PM UTC-4, Gianguido Cianci wrote:
> Hi all,
>
>
>
> I am wondering if anybody has suggestions on how to improve the function below. It seems ok

for floating precision numbers.
>
>
>
> For integers it's a different story:
>
> It works great if N<<TOTAL. When N approaches TOTAL I get a few numbers and then a bunch of zeros... Also, setting /DIFFERENT makes it run for ever if N is large. Also, the sum of res adds up TOTAL=/-1, not always to TOTAL exactly...
>
>
>
> Suggestions?
>
>
>
> Thanks,
>
> Gianguido
>
>
>
>
>
>
>
> FUNCTION nrndaddto, n, total, integers = integers, different = different
>
>
>
> compile_opt idl2
>
>
>
> res = dblarr(n)
>
> res[0] = randomu(seed, 1, /double)*(total)
>
>
>
> FOR i = 1, n-2 DO BEGIN
>
> res[i] = randomu(seed, 1, /double)*(total-total(res[0:i-1], /double))
>
> ENDFOR
>
> res[n-1] = total-total(res[0:n-2], /double)
>

>
>
> IF ~keyword_set(integers) THEN integers = 0
>
>
>
> IF keyword_set(integers) THEN res = round(res)
>
> IF keyword_set(different) THEN BEGIN
>
> IF n_elements(res) NE n_elements(unique(res, /sort)) THEN res = $
>
> nrndaddto(n, total, integers = integers, different = 1)
>
> ENDIF
>
>
>
>
>
> RETURN, res
>
> END

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Russell Ryan on Thu, 07 Aug 2014 14:08:03 GMT
View Forum Message <> Reply to Message

On Wednesday, August 6, 2014 11:52:47 PM UTC-4, Gianguido Cianci wrote:
> Hi all,
>
>
>
> I am wondering if anybody has suggestions on how to improve the function below. It seems ok for floating precision numbers.
>
>
>
> For integers it's a different story:
>
> It works great if N<<TOTAL. When N approaches TOTAL I get a few numbers and then a bunch of zeros... Also, setting /DIFFERENT makes it run for ever if N is large. Also, the sum of res adds up TOTAL=/-1, not always to TOTAL exactly...
>
>
>
> Suggestions?

>
>
>
> Thanks,
>
> Gianguido
>
>
>
>
>
>
>
> FUNCTION nrndaddto, n, total, integers = integers, different = different
>
>
>
> compile_opt idl2
>
>
>
> res = dblarr(n)
>
> res[0] = randomu(seed, 1, /double)*(total)
>
>
>
> FOR i = 1, n-2 DO BEGIN
>
>   res[i] = randomu(seed, 1, /double)*(total-total(res[0:i-1], /double))
>
> ENDFOR
>
> res[n-1] = total-total(res[0:n-2], /double)
>
>
>
> IF ~keyword_set(integers) THEN integers = 0
>
>
>
> IF keyword_set(integers) THEN res = round(res)
>
> IF keyword_set(different) THEN BEGIN
>
>   IF n_elements(res) NE n_elements(unique(res, /sort)) THEN res = $
>
>   nrndaddto(n, total, integers = integers, different = 1)

>
> ENDIF
>
>
>
>
>
> RETURN, res
>
> END

I should've said. Since the running total is converging to the requested total, the numbers are getting smaller with time. That is almost certainly related to why your sequence doesn't seem uniform.

R

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Russell Ryan on Thu, 07 Aug 2014 14:18:26 GMT
View Forum Message <> Reply to Message

You might have better luck with:

n_rand_var = 1000
requested_total=1000.

rand=randomu(seed,n_rand_var,/double)
rand*=(requested_total/total(rand))

This looks to be uniformly distributed, however it's not clear over what range it's uniform because the total(rand) in the denominator isn't necessarily the same. In the limit of n_rand_var -> infinity, then I think the total will converge to n_rand_var/2 and so the range will be

2*requested_total/n_rand_var

but for n_rand_var != infinity then it's a bit vague.

R

On Wednesday, August 6, 2014 11:52:47 PM UTC-4, Gianguido Cianci wrote:
> Hi all,
>
>
>

> I am wondering if anybody has suggestions on how to improve the function below. It seems ok for floating precision numbers.
>
>
>
> For integers it's a different story:
>
> It works great if N<<TOTAL. When N approaches TOTAL I get a few numbers and then a bunch of zeros... Also, setting /DIFFERENT makes it run for ever if N is large. Also, the sum of res adds up TOTAL=/-1, not always to TOTAL exactly...
>
>
>
> Suggestions?
>
>
>
> Thanks,
>
> Gianguido
>
>
>
>
>
>
>
> FUNCTION nrndaddto, n, total, integers = integers, different = different
>
>
>
> compile_opt idl2
>
>
>
> res = dblarr(n)
>
> res[0] = randomu(seed, 1, /double)*(total)
>
>
>
> FOR i = 1, n-2 DO BEGIN
>
> res[i] = randomu(seed, 1, /double)*(total-total(res[0:i-1], /double))
>
> ENDFOR
>
> res[n-1] = total-total(res[0:n-2], /double)

```
>
>
>
>  IF ~keyword_set(integers) THEN integers = 0
>
>
>
>  IF keyword_set(integers) THEN res = round(res)
>
>  IF keyword_set(different) THEN BEGIN
>
>    IF n_elements(res) NE n_elements(unique(res,  /sort)) THEN res = $
>
>    nrndaddto(n, total, integers = integers, different = 1)
>
>  ENDIF
>
>
>
>
>
>  RETURN,  res
>
>  END
```

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Michael Galloy on Fri, 08 Aug 2014 16:54:19 GMT
View Forum Message <> Reply to Message

On 8/6/14, 9:52 PM, Gianguido Cianci wrote:
> Hi all,
>
> I am wondering if anybody has suggestions on how to improve the function below. It seems ok for floating precision numbers.
>
> For integers it's a different story:
> It works great if N<<TOTAL. When N approaches TOTAL I get a few numbers and then a bunch of zeros... Also, setting /DIFFERENT makes it run for ever if N is large. Also, the sum of res adds up TOTAL=/-1, not always to TOTAL exactly...
>
> Suggestions?
>
> Thanks,
> Gianguido
>
>
>

```
>  FUNCTION nrndaddto,  n,  total,  integers = integers,  different = different
>
>  compile_opt idl2
>
>  res = dblarr(n)
>  res[0] = randomu(seed, 1,  /double)*(total)
>
>  FOR i = 1,  n-2 DO BEGIN
>     res[i] = randomu(seed, 1,  /double)*(total-total(res[0:i-1],  /double))
>  ENDFOR
>  res[n-1] = total-total(res[0:n-2],  /double)
>
>  IF ~keyword_set(integers) THEN integers = 0
>
>  IF keyword_set(integers) THEN res = round(res)
>  IF keyword_set(different) THEN BEGIN
>     IF n_elements(res) NE n_elements(unique(res,  /sort)) THEN res = $
>     nrndaddto(n, total, integers = integers, different = 1)
>  ENDIF
>
>
>  RETURN,  res
>  END
>
```

What about just FLOOR the normalized float values and then just
increment the required number of values with the largest remainders?

```
function mg_random_to_total, n, sum, seed=seed
  compile_opt strictarr

  x = randomu(seed, n)
  x *= sum / total(x, /preserve_type)
  int_x = long(floor(x))
  dec_x = x - int_x

  int_total = total(int_x, /preserve_type)
  ind = sort(dec_x)
  int_x[ind[0:(sum - int_total - 1)]]++

  return, int_x
end
```

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)

Research Mathematician
Tech-X Corporation

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Russell Ryan on Fri, 08 Aug 2014 19:32:47 GMT
View Forum Message <> Reply to Message

Hi Mike,
I might be interested in GPULib. I see it's quite pricey (at least for my budget) and Tech-X offers a free trial. But before I go through the trouble of even getting the free trial, what else can you tell me about GPULib?

Specifically, I was wondering about what hardware/software do I need to use GPULib? I use Mac OSX 10.8.5 and IDL 8.2.3 at present, and that sounded sufficient. But anything else I should be aware of? Such as GPU cards, RAM, etc.?

Can you give any examples of the code usage? Like what will my IDL code now look like?

What about if I need to port the code to another workstation?

Anything else a GPU newbie (but seasoned IDLer) should know or should ask?

THanks,
Russell

On Friday, August 8, 2014 12:54:19 PM UTC-4, Mike Galloy wrote:
> On 8/6/14, 9:52 PM, Gianguido Cianci wrote:
>
>> Hi all,
>
>>
>
>> I am wondering if anybody has suggestions on how to improve the function below. It seems ok for floating precision numbers.
>
>>
>
>> For integers it's a different story:
>
>> It works great if N<<TOTAL. When N approaches TOTAL I get a few numbers and then a bunch of zeros... Also, setting /DIFFERENT makes it run for ever if N is large. Also, the sum of res adds up TOTAL=/-1, not always to TOTAL exactly...
>
>>
>
>> Suggestions?

>
>>
>
>> Thanks,
>
>> Gianguido
>
>>
>
>>
>
>>
>
>> FUNCTION nrndaddto, n, total, integers = integers, different = different
>
>>
>
>> compile_opt idl2
>
>>
>
>> res = dblarr(n)
>
>> res[0] = randomu(seed, 1, /double)*(total)
>
>>
>
>> FOR i = 1, n-2 DO BEGIN
>
>>   res[i] = randomu(seed, 1, /double)*(total-total(res[0:i-1], /double))
>
>> ENDFOR
>
>> res[n-1] = total-total(res[0:n-2], /double)
>
>>
>
>> IF ~keyword_set(integers) THEN integers = 0
>
>>
>
>> IF keyword_set(integers) THEN res = round(res)
>
>> IF keyword_set(different) THEN BEGIN
>
>>   IF n_elements(res) NE n_elements(unique(res, /sort)) THEN res = $
>
>>   nrndaddto(n, total, integers = integers, different = 1)

```
>
>>  ENDIF
>
>>
>
>>
>
>>  RETURN,  res
>
>>  END
>
>>
>
>
>
>  What about just FLOOR the normalized float values and then just
>
>  increment the required number of values with the largest remainders?
>
>
>
>  function mg_random_to_total, n, sum, seed=seed
>
>     compile_opt strictarr
>
>
>
>     x = randomu(seed, n)
>
>     x *= sum / total(x, /preserve_type)
>
>     int_x = long(floor(x))
>
>     dec_x = x - int_x
>
>
>
>     int_total = total(int_x, /preserve_type)
>
>     ind = sort(dec_x)
>
>     int_x[ind[0:(sum - int_total - 1)]]++
>
>
>
>     return, int_x
>
>  end
```

>
>
>
> Mike
>
> --
>
> Michael Galloy
>
> www.michaelgalloy.com
>
> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
>
> Research Mathematician
>
> Tech-X Corporation

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Michael Galloy on Fri, 08 Aug 2014 22:00:03 GMT
View Forum Message <> Reply to Message

On 8/8/14, 1:32 PM, rryan@stsci.edu wrote:
> Hi Mike, I might be interested in GPULib. I see it's quite pricey
> (at least for my budget) and Tech-X offers a free trial. But before
> I go through the trouble of even getting the free trial, what else
> can you tell me about GPULib?

A couple of resources:

* documentation for GPULib routines:
http://www.txcorp.com/images/docs/gpulib/1.6.2/html/index.ht ml

* I write about GPULib on my website occasionally, see:

michaelgalloy.com/index.php?s=gpulib&submit=Search

* The official blog is at hgpulib.blogspot.com

> Specifically, I was wondering about what hardware/software do I need
> to use GPULib? I use Mac OSX 10.8.5 and IDL 8.2.3 at present, and
> that sounded sufficient. But anything else I should be aware of?
> Such as GPU cards, RAM, etc.?

Currently, you absolutely need to have CUDA-enabled GPU (any modern NVIDIA graphics card). The better the card, the better the performance. Most laptop GPUs can get 2-5x speedup on our demos, while top-end GPUs can get 40x or better speedups.

For software, IDL 8.2 and CUDA 5.0 on OS X (10.7+), Windows (7, Server 2008), or Linux (CentOS5, CentOS6, RedHat Enterprise Linux 5, Fedora 16). If your software doesn't quite match up, I can usually make a custom build for you.

> Can you give any examples of the code usage? Like what will my IDL
> code now look like?

It could be as simple as:

```
gpuinit
dx = gpuFindgen(10)
dy = gpuFindgen(10)
dz = dx + dy
```

That last line could also be done this way:

```
dz = gpuFltarr(10)
dz = gpuAdd(dx, dy, LHS=dz)
```

which can be more efficient in certain situations.

There are basically a bunch of routines with the "gpu" prefix that have a similar interface as the normal IDL library routine, but take GPU variables instead of normal ones. See the API documentation link I gave above for a list of routines available.

There are also several demos in the trial that you can see speedups and browse example code.

> What about if I need to port the code to another workstation?

Should be fine (no modification) as long as the new workstation also meets the requirements above.

> Anything else a GPU newbie (but seasoned IDLer) should know or should
> ask?

Not that I can think of, but feel free to ask if you have any more questions!

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by cgguido on Mon, 11 Aug 2014 21:14:01 GMT
View Forum Message <> Reply to Message

Thanks everyone for your replies. Will have a closer look next week, once I'm back States-side.

G

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by markb77 on Mon, 11 Aug 2014 22:37:08 GMT
View Forum Message <> Reply to Message

On Saturday, August 9, 2014 12:00:03 AM UTC+2, Mike Galloy wrote:
> On 8/8/14, 1:32 PM, rryan@stsci.edu wrote:
>
>> Hi Mike, I might be interested in GPULib. I see it's quite pricey
>
>> (at least for my budget) and Tech-X offers a free trial. But before
>
>> I go through the trouble of even getting the free trial, what else
>
>> can you tell me about GPULib?
>
>
>
> A couple of resources:
>
>
>
> * documentation for GPULib routines:
>
> http://www.txcorp.com/images/docs/gpulib/1.6.2/html/index.ht ml
>
>
>
> * I write about GPULib on my website occasionally, see:
>
>
>
> michaelgalloy.com/index.php?s=gpulib&submit=Search
>
>
>

>   * The official blog is at hgpulib.blogspot.com
>
>
>
>> Specifically, I was wondering about what hardware/software do I need
>
>> to use GPULib? I use Mac OSX 10.8.5 and IDL 8.2.3 at present, and
>
>> that sounded sufficient. But anything else I should be aware of?
>
>> Such as GPU cards, RAM, etc.?
>
>
>
> Currently, you absolutely need to have CUDA-enabled GPU (any modern
>
> NVIDIA graphics card). The better the card, the better the performance.
>
> Most laptop GPUs can get 2-5x speedup on our demos, while top-end GPUs
>
> can get 40x or better speedups.
>
>
>
> For software, IDL 8.2 and CUDA 5.0 on OS X (10.7+), Windows (7, Server
>
> 2008), or Linux (CentOS5, CentOS6, RedHat Enterprise Linux 5, Fedora
>
> 16). If your software doesn't quite match up, I can usually make a
>
> custom build for you.
>
>
>
>> Can you give any examples of the code usage? Like what will my IDL
>
>> code now look like?
>
>
>
> It could be as simple as:
>
>
>
> gpuinit
>
> dx = gpuFindgen(10)
>

> dy = gpuFindgen(10)
>
> dz = dx + dy
>
>
>
> That last line could also be done this way:
>
>
>
> dz = gpuFltarr(10)
>
> dz = gpuAdd(dx, dy, LHS=dz)
>
>
>
> which can be more efficient in certain situations.
>
>
>
> There are basically a bunch of routines with the "gpu" prefix that have
>
> a similar interface as the normal IDL library routine, but take GPU
>
> variables instead of normal ones. See the API documentation link I gave
>
> above for a list of routines available.
>
>
>
> There are also several demos in the trial that you can see speedups and
>
> browse example code.
>
>
>
>> What about if I need to port the code to another workstation?
>
>
>
> Should be fine (no modification) as long as the new workstation also
>
> meets the requirements above.
>
>
>
>> Anything else a GPU newbie (but seasoned IDLer) should know or should
>

>> ask?
>
>
>
> Not that I can think of, but feel free to ask if you have any more
>
> questions!
>
>
>
> Mike
>
> --
>
> Michael Galloy
>
> www.michaelgalloy.com
>
> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
>
> Research Mathematician
>
> Tech-X Corporation

hi Mike,

A while back you were working on some Levenberg-Marquardt curve fitting examples using GPULIB. Are those ready to be made public?

thanks,
Mark

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Michael Galloy on Tue, 12 Aug 2014 21:18:44 GMT
View Forum Message <> Reply to Message

On 8/11/14, 4:37 PM, superchromix wrote:
> A while back you were working on some Levenberg-Marquardt curve
> fitting examples using GPULIB. Are those ready to be made public?

Not yet. I hope to have a summer release to update IDL/CUDA and a few bug fixes, but I'm not sure if the curve fitting stuff will get in there as well.

Mike
--

Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by markb77 on Wed, 13 Aug 2014 07:43:44 GMT
View Forum Message <> Reply to Message

On Tuesday, August 12, 2014 11:18:44 PM UTC+2, Mike Galloy wrote:
> On 8/11/14, 4:37 PM, superchromix wrote:
>
>> A while back you were working on some Levenberg-Marquardt curve
>
>> fitting examples using GPULIB. Are those ready to be made public?
>
>
>
> Not yet. I hope to have a summer release to update IDL/CUDA and a few
>
> bug fixes, but I'm not sure if the curve fitting stuff will get in there
>
> as well.
>
>
>
> Mike
>
> --
>
> Michael Galloy
>
> www.michaelgalloy.com
>
> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
>
> Research Mathematician
>
> Tech-X Corporation

ok, thanks for the update.

have you seen this? It claims to be a CUDA implementation of MPfit:

Zhu X, Zhang D (2013) Efficient Parallel Levenberg-Marquardt Model Fitting towards Real-Time Automated Parametric Imaging Microscopy. PLoS ONE 8(10): e76665.

doi:10.1371/journal.pone.0076665

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Michael Galloy on Wed, 13 Aug 2014 20:37:00 GMT
View Forum Message <> Reply to Message

On 8/13/14, 1:43 AM, superchromix wrote:
> On Tuesday, August 12, 2014 11:18:44 PM UTC+2, Mike Galloy wrote:
>> On 8/11/14, 4:37 PM, superchromix wrote:
>>
>>> A while back you were working on some Levenberg-Marquardt curve
>>
>>> fitting examples using GPULIB. Are those ready to be made
>>> public?
>>
>>
>>
>> Not yet. I hope to have a summer release to update IDL/CUDA and a
>> few
>>
>> bug fixes, but I'm not sure if the curve fitting stuff will get in
>> there
>>
>> as well.
>>
>>
>>
>> Mike
>>
>> --
>>
>> Michael Galloy
>>
>> www.michaelgalloy.com
>>
>> Modern IDL: A Guide to IDL Programming
>> (http://modernidl.idldev.com)
>>
>> Research Mathematician
>>
>> Tech-X Corporation
>
> ok, thanks for the update.
>
> have you seen this? It claims to be a CUDA implementation of MPfit:
>
> Zhu X, Zhang D (2013) Efficient Parallel Levenberg-Marquardt Model

> Fitting towards Real-Time Automated Parametric Imaging Microscopy.
> PLoS ONE 8(10): e76665. doi:10.1371/journal.pone.0076665
>

Yes, but I am hoping to keep the implementation as "IDL native" as possible for more flexibility. Craig's MPFIT is a cited reference for this paper.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Craig Markwardt on Wed, 13 Aug 2014 23:16:04 GMT
View Forum Message <> Reply to Message

On Wednesday, August 13, 2014 3:43:44 AM UTC-4, superchromix wrote:

> ok, thanks for the update.
>
> have you seen this? It claims to be a CUDA implementation of MPfit:
>
> Zhu X, Zhang D (2013) Efficient Parallel Levenberg-Marquardt Model Fitting towards Real-Time Automated Parametric Imaging Microscopy. PLoS ONE 8(10): e76665. doi:10.1371/journal.pone.0076665

These authors did contact me. Although I haven't used their code, they do seem very capable and I applaud their effort!

Craig

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by markb77 on Thu, 14 Aug 2014 10:10:10 GMT
View Forum Message <> Reply to Message

On Wednesday, August 13, 2014 10:37:00 PM UTC+2, Mike Galloy wrote:
> On 8/13/14, 1:43 AM, superchromix wrote:
>
>> On Tuesday, August 12, 2014 11:18:44 PM UTC+2, Mike Galloy wrote:
>
>>> On 8/11/14, 4:37 PM, superchromix wrote:

>
>>>
>
>>>> A while back you were working on some Levenberg-Marquardt curve
>
>>>
>
>>>> fitting examples using GPULIB. Are those ready to be made
>
>>>> public?
>
>>>
>
>>>
>
>>>
>
>>>
>
>>> Not yet. I hope to have a summer release to update IDL/CUDA and a
>
>>> few
>
>>>
>
>>> bug fixes, but I'm not sure if the curve fitting stuff will get in
>
>>> there
>
>>>
>
>>> as well.
>
>>>
>
>>>
>
>>>
>
>>>
>
>>> Mike
>
>>>
>
>>> --
>
>>>
>
>>> Michael Galloy
>
>>>

>
>>> www.michaelgalloy.com
>
>>>
>
>>> Modern IDL: A Guide to IDL Programming
>
>>> (http://modernidl.idldev.com)
>
>>>
>
>>> Research Mathematician
>
>>>
>
>>> Tech-X Corporation
>
>>
>
>> ok, thanks for the update.
>
>>
>
>> have you seen this? It claims to be a CUDA implementation of MPfit:
>
>>
>
>> Zhu X, Zhang D (2013) Efficient Parallel Levenberg-Marquardt Model
>
>> Fitting towards Real-Time Automated Parametric Imaging Microscopy.
>
>> PLoS ONE 8(10): e76665. doi:10.1371/journal.pone.0076665
>
>>
>
>
>
> Yes, but I am hoping to keep the implementation as "IDL native" as
>
> possible for more flexibility. Craig's MPFIT is a cited reference for
>
> this paper.
>
>
>
> Mike
>
> --

>
> Michael Galloy
>
> www.michaelgalloy.com
>
> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
>
> Research Mathematician
>
> Tech-X Corporation

I wonder... how easy / difficult it would be to take their CUDA code and run it with GPUlib as a "custom kernel" ?

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Michael Galloy on Thu, 14 Aug 2014 14:25:30 GMT
View Forum Message <> Reply to Message

On 8/14/14, 4:10 am, superchromix wrote:
> I wonder... how easy / difficult it would be to take their CUDA code
> and run it with GPUlib as a "custom kernel" ?

I did not see where their source was available. We would need .cu or
.ptx (compiled .cu) files to run as a custom kernel.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by markb77 on Thu, 14 Aug 2014 14:55:02 GMT
View Forum Message <> Reply to Message

On Thursday, August 14, 2014 4:25:30 PM UTC+2, Mike Galloy wrote:
> On 8/14/14, 4:10 am, superchromix wrote:
>
>> I wonder... how easy / difficult it would be to take their CUDA code
>
>> and run it with GPUlib as a "custom kernel" ?
>

>
>
> I did not see where their source was available. We would need .cu or
>
> .ptx (compiled .cu) files to run as a custom kernel.
>
>
>
> Mike
>
> --
>
> Michael Galloy
>
> www.michaelgalloy.com
>
> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
>
> Research Mathematician
>
> Tech-X Corporation

The source code looks to be all there. Look in the Supporting Information section - it's called "File_S1.zip". There are .cu and .cuh files, along with some matlab stuff.

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by markb77 on Thu, 14 Aug 2014 14:55:25 GMT
View Forum Message <> Reply to Message

On Thursday, August 14, 2014 4:55:02 PM UTC+2, superchromix wrote:
> On Thursday, August 14, 2014 4:25:30 PM UTC+2, Mike Galloy wrote:
>
>> On 8/14/14, 4:10 am, superchromix wrote:
>
>>
>
>>> I wonder... how easy / difficult it would be to take their CUDA code
>
>>
>
>>> and run it with GPUlib as a "custom kernel" ?
>
>>
>
>>
>

>>
>
>> I did not see where their source was available. We would need .cu or
>
>>
>
>> .ptx (compiled .cu) files to run as a custom kernel.
>
>>
>
>>
>
>>
>
>> Mike
>
>>
>
>> --
>
>>
>
>> Michael Galloy
>
>>
>
>> www.michaelgalloy.com
>
>>
>
>> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
>
>>
>
>> Research Mathematician
>
>>
>
>> Tech-X Corporation
>
>
>
>
>
>
> The source code looks to be all there. Look in the Supporting Information section - it's called "File_S1.zip". There are .cu and .cuh files, along with some matlab stuff.

link: http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjourna

l.pone.0076665#pone.0076665.s001

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Michael Galloy on Mon, 18 Aug 2014 21:21:30 GMT
View Forum Message <> Reply to Message

On 8/14/14, 8:55 AM, superchromix wrote:
>> The source code looks to be all there. Look in the Supporting
>> Information section - it's called "File_S1.zip". There are .cu and
>> .cuh files, along with some matlab stuff.
>
> link:
> http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjourna
l.pone.0076665#pone.0076665.s001

They have a lot of source code, but not for the main fitting routine
GPU_LMFit -- they just provide pre-compiled .lib files for Windows.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Michael Galloy on Tue, 19 Aug 2014 17:03:08 GMT
View Forum Message <> Reply to Message

On 8/18/14, 3:21 PM, Michael Galloy wrote:
> On 8/14/14, 8:55 AM, superchromix wrote:
>>> The source code looks to be all there. Look in the Supporting
>>> Information section - it's called "File_S1.zip". There are .cu
>>> and .cuh files, along with some matlab stuff.
>>
>> link:
>> http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjourna
l.pone.0076665#pone.0076665.s001
>>
>>
>
> They have a lot of source code, but not for the main fitting routine
> GPU_LMFit -- they just provide pre-compiled .lib files for Windows.
>

> Mike

I contacted the author about the source code and, unfortunately, they do
not make it available. The compiled version is only available for Windows.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation
