GammaLib - Feature #1407

Support the column "PULSE_PHASE" in GCTAEventList

01/16/2015 09:06 AM - Mayer Michael

Status:	Closed	**Start date:**	01/16/2015
Priority:	Normal	**Due date:**	
Assigned To:	Mayer Michael	**% Done:**	100%
Category:		**Estimated time:**	0.50 hour
Target version:	1.0.0		

Description

With the advent of HESS II, and certainly when CTA comes online, pulsars are an important research topic.
Currently GCTAEventList does not include the possibility to have a pulse phase for each event (analogous to Fermi-LAT stored as "PULSE_PHASE"). Especially when running ctselect, the pulse phase column gets lost when saving the new event file.
I think we could add this as an optional column, including a boolean m_has_phase to GCTAEventList (just that it gets loaded and saved). Probably, we might also need to add a protected member m_phase to the GEvent-class, too?

History

#1 - 01/16/2015 10:02 AM - Knödlseder Jürgen

I agree that this should be supported.

I won't add a m_phase member to the GEvent class, which does in fact not have any data members. Data members should be implemented at the derived class level to guarantee maximum flexibility (and reduce memory usage for cases where this information is not needed). The m_phase member needs to be added at the GCTAEventAtom level.

Globally, I'm still looking for a clever technique that allows for flexibility in GCTAEventAtom concerning the loaded information. Having static members in that class is not very nice as it implies quite some burden on the memory (one needs the load all data, even if one only want to access for example the event times).

But for the moment, working with a flag (or a specific value such as NAN to check whether the information exists) should be okay.

#2 - 01/19/2015 03:14 PM - Mayer Michael

- Status changed from New to Pull request

- Assigned To set to Mayer Michael

- Target version set to 1.0.0

- % Done changed from 0 to 100

- Estimated time set to 0.50

> I won't add a m_phase member to the GEvent class, which does in fact not have any data members. Data members should be implemented at the derived class level to guarantee maximum flexibility (and reduce memory usage for cases where this information is not needed). The m_phase member needs to be added at the GCTAEventAtom level.

Agree.

> Globally, I'm still looking for a clever technique that allows for flexibility in GCTAEventAtom concerning the loaded information. Having static members in that class is not very nice as it implies quite some burden on the memory (one needs the load all data, even if one only want to access for example the event times).

I don't know if it would be a memory reduction to use vectors of event properties instead of static members. Querying for a certain property would then imply to use a specific string, e.g. event->get("PULSE_PHASE") or event->get("ENERGY"). I guess however, that there would be a lot of string comparisons which might slow down the computation. Maybe we could still use that for all parameters except energy, direction and time (which are the most fundamental ones).

> But for the moment, working with a flag (or a specific value such as NAN to check whether the information exists) should be okay.

I implemented a flag m_has_phase to GCTAEventList and the float value m_phase to GCTAEventAtom (in the long run, we might have to check if float is precise enough). The pulse phase is only written to a FITS file if m_has_phase=true, i.e. the column "PULSE_PHASE" was already present while reading. I ran all unit tests and did some quick checks with ctselect. Everything looks fine with these changes. The code is available on *1407-support-pulse-phase-for-GCTAEventList*

#3 - 01/30/2015 09:55 PM - Knödlseder Jürgen

Mayer Michael wrote:

> I don't know if it would be a memory reduction to use vectors of event properties instead of static members. Querying for a certain property would then imply to use a specific string, e.g. event->get("PULSE_PHASE") or event->get("ENERGY"). I guess however, that there would be a lot of string comparisons which might slow down the computation. Maybe we could still use that for all parameters except energy, direction and time (which are the most fundamental ones).

Yes, I would be very worried about computing time, but maybe this could be used for "additional" parameters. One then would need however methods such as real(), integer() and string() to deal with different content types (not every parameter is a float). There is still the problem that when you read in a file for example using GCTAObservation you'd like that it writes out the same file when you call write. However, if not all data is loaded it cannot do that. This is the other part I'm still thinking about how to deal with.

#4 - 01/30/2015 10:07 PM - Knödlseder Jürgen

Mayer Michael wrote:

> I implemented a flag m_has_phase to GCTAEventList and the float value m_phase to GCTAEventAtom (in the long run, we might have to check if float is precise enough). The pulse phase is only written to a FITS file if m_has_phase=true, i.e. the column "PULSE_PHASE" was already present while reading. I ran all unit tests and did some quick checks with ctselect. Everything looks fine with these changes. The code is available on *1407-support-pulse-phase-for-GCTAEventList*

I looked into the code. Looks fine. The only modification I propose is to rename "PULSE_PHASE" to "PHASE" to make this more generic. One could also think about using this for a binary orbital phase, for example. I also changed the variable names accordingly.

#5 - 01/30/2015 10:35 PM - Knödlseder Jürgen

- Status changed from Pull request to Closed

Merged into devel.

#6 - 02/02/2015 11:34 AM - Mayer Michael

> The only modification I propose is to rename "PULSE_PHASE" to "PHASE" to make this more generic. One could also think about using this for a binary orbital phase, for example. I also changed the variable names accordingly.

The only reason to have "PULSE_PHASE" was to be conform with Fermi-LAT. But I agree that "PHASE" is more generic.
Currently, the PHASE column is added while exporting the IACT data to FITS. However, I guess ultimately, we might have ctpphase which can add whatever phases to the event lists. Accordingly, we can define the column names ourselves later. \

> Merged into devel.

Thanks.

#7 - 02/02/2015 01:15 PM - Knödlseder Jürgen

I had in fact two tools planned:

- one that attributes pulsar phases based on pulsar ephemerides
- one that attributes binary phases based on binary ephemerides
  The difference is that pulsar ephemerides need Barycentering, and one may want to use tools such as tempo2, while binary orbits don't need Barycentering and they probably have input parameters very different to pulsars.