

2024-7-9 Meeting notes

Date

Jul 9, 2024 , 10:00 AM EST (US)

Participants

- @Scot Breitenfeld ★
- @Vicky Moschou
- Scott Imlay ★
- @David Gutzwiller
- @Simone Crippa
- @Tobias Leicht
- @Gregory Sjaardema ★
- @David Gutzwiller
- @Stephen Guzik ★
- @dmitry.s.kamenetskiy
- @Tony Garratt
- @Patrick Baker (Has new representative)
- @Earl P Duque
- @ZJ Wang
- @Matthias Möller
- Kamenetskiy, Dmitry S
- @Mickael PHILIT ★
- @Marc Poinot
- @Koen Hillewaert
- Giorgos Strofylas (BETA CAE)
- Stavros Arestis (BETA CAE)
- David Garlich ★
- Mark Lohry ★
- Julien Coulet ★

★ In attendance

Guest representatives from *convergecdf*

Steering Committee Issues

New committee representatives:


@Christopher Rumsey is being replaced by Mark Lohry for NASA.

@Pierre-Jacques Legay is being replaced by Julien Coulet for ONERA.

Approved new representatives.

Discussion topics

Item	Presenter	Notes
Approve 2024/4/9 minutes.	@Scot Breitenfeld	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Post the minutes of the last meeting on the CGNS webpage. Approved
cgns.org	@Scot Breitenfeld	Thomas Hauser will add @Scot Breitenfeld as an alternative owner. There are no updates; I've reached out to Thomas.
high-level editing tools for the documentation page	@Marc Poinot @Christopher Rumsey @Scot Breitenfeld	The Doxygen branch was merged to develop. Started to explore using Breathe and Sphinx using the <i>develop</i> branch. UPDATE: Still in progress.
Converting CGNSTalk archive to Github Discussion Posts	@Scot Breitenfeld	No update.
Discussion of the need to extend the standard to include identifiers and references	@Scot Breitenfeld	With more widespread sharing of CGNS data, such as ML training, it might be helpful to standardize unique identifiers (global) for the file (maybe data, local?). Some examples: <ul style="list-style-type: none"> • Online Resources: creator URL, publisher URL, contributors • UUIDs, date convention, citations • Descriptive keywords • Lineage: audit trail for mods to original data @Scot Breitenfeld Submitted a DOE pre-proposal to gauge interest in funding the work. No feedback from DOE.
mixed element/face zones discussion	@David Garlisch	<p>ParaView: Mixed definition of unstructured zone by elements and by faces is not valid · CGNS CGNS · Discussion #720</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> @David Garlisch will post to cgnstalk files that show the issue, @Gregory Sjaardema will look into it. Unrelated, the files also cause Paraview to crash. <ul style="list-style-type: none"> <input type="checkbox"/> @Scot Breitenfeld will follow-up with @Gregory Sjaardema <input type="checkbox"/> @Vicky Moschou and others will draft a document to discuss hardening the specification for faces in the case of mixed meshes. <input type="checkbox"/> @Scot Breitenfeld will look into what to do next.
CPEX #0046 – CPEX 0046 - New p article extension proposal · Issue #772 · CGNS/CGNS	converge cfd	Discussion of CPEX #0047, CPEX0046-v2-rev1-2.pdf Tobias feedback: However, I carefully read the CPEX (thanks for sending it well ahead of time), and here is my feedback and opinion. <ul style="list-style-type: none"> • The impact of the proposed changes on existing functionality and interfaces is minimal, so I don't have (strong) objections for accepting this if a majority is in favor. • I know too little about particle methods to judge on whether the various lists of names (methods, models, new variables) are useful / exhaustive, but I trust the authors (and community) on this – and in any case this can always be adapted later on. • For me, the relevant question is already part of the document "If the <i>ParticleZone_t</i> is so similar to a <i>Zone_t</i>, why create a new node? Zones are dedicated to holding a simulation mesh. As in, elements connected to each other. Each particle is independent of other particles. It has no neighbors." I have <i>not</i> been the one asking that original question (in fact I was unaware of it until now). But I would strongly second that question. And I would challenge the validity (or maybe rather the relevance) of the reply offered in the document. In fact, I have been working in a software environment with "relaxed requirements" on what a mesh is for a while. In particular, having just nodes and no elements is a valid option for the unstructured mesh path in that software – and a frequently used case, in fact. Admittedly it takes newcomers (including myself at the time ;) a while to mentally accept that concept, but once that is done it really is a helpful tool that allows you to conveniently do things that would otherwise require special casing and extra effort. In fact it seems that storing particle data would require no changes whatsoever once we accept nodes in a zone without additional element connectivity to represent particles – standard nodal data can be used then. (Unless I miss something relevant, of course.) It might still be useful to have

		<p>new standardized quantities and model names – but that is a much simpler question. In that sense I would like to motivate the authors of the CPEX and the CGNS Steering Committee to reconsider this idea in the upcoming discussion. (Keeping in mind that – as I said above – this does <i>_not_</i> constitute a veto from my side.)</p> <p>Mickael:</p> <p>To complete your point concerning zone_t vs particlezone_t, I think in the case of zone_t we would need a new zonetype (let's say 'Particle') to clearly differentiate from structured and unstructured existing mesh case. This information is currently stored under Zone_t node which means parser need to enter the hdf5 node group of the zone before identifying that it's not a 'standard' mesh. The question is thus are we ready to pay this price? Should we modify the Zone_t mapping to get this kind of information at the node lever.</p> <p>Alexandre:</p> <p>I wanted to provide a detailed answer to what Tobias sent. I meant to send this response a week ago, but I now realize the email did not go out. I hope this will reach you before our meeting.</p> <p>Indeed, using the zone with little or no modifications to hold particle data is the main topic that should be discussed. In fact, this is what we first tried (see How to export Lagrangian data - CGNS CGNS - Discussion #682). We found two sticking points for this: The CellDimension and the fact that it does not make it clear that we are exporting particle data.</p> <p>The first issue is quite simple, but unfortunately blocks us from the start: the CellDimension is defined at the base level and inherited by the zone as a parameter. Many users, like us, will be exporting both 3D cells holding Eulerian data and 1D particle data. The base would need 3 as a CellDimension to satisfy the need of the zone holding Eulerian data, which would be incorrect for the zone holding particle data. We really do not like the idea of putting particle data and Eulerian data from a single simulation in two different bases. If we were to do this though, it is true that particles could be defined as NODE elements in an unstructured zone. Technically though, particles are not NODE elements, as they are not connected to each other. We also fear that this way of storing particle data is quite obscure.</p> <p>The purpose of this standard is to store data in a format that can be accessed by all without any ambiguity. We fear that using the base and zone to hold particle data would lead us to encode particle results using existing nodes and lose the meaning associated with the data. If a user wanted to analyze the particle data, we would have to explain to them that they need to look for the NODE data and treat those in a special way. We believe that this goes against the principle of the standard. To put it another way: We are convinced that anyone on this comity could implement a particle data analysis tool based on the documentation and MLL we submitted alone, without the need for an example case. And that's what the standard is all about.</p> <p>As Mickael is suggesting, part of this issue could be solved by introducing a new ZoneType: Particles. This would make it explicit that the point cloud defined in the zone is a set of particles. We could write a rule saying that CellDimension is only inherited by structured and unstructured zones. We would have to explicitly say that particles should be defined using GridCoordinates only and not elements, as the vertices implicitly define cells. We would explicitly say that since cells are implicitly defined by vertices, the GridLocation is Vertex and not CellCenter. This would imply that CellSize becomes optional which is a breaking change. Since CellSize is only needed to define meshes and other Eulerian nodes like the DiffusionModel, we might be able to get away with setting it to 0 for ZoneType particles. These are the sort of details and confusions we were alluding to when we said the "zones are dedicated to holding a simulation mesh". These sorts of changes will in practice break many readers. Virtually all readers currently expect zones to be either structured or unstructured. As Mickael points out, readers would now need to read the zone type first to at least ignore ZoneType Particle if they cannot read particles.</p> <p>I would also like to raise the following question, not as a counter argument, but really to get the opinion of the community: If particles can be stored using the current standard, why has it not been done yet when it is such a common request? What difficulties have others run into?</p> <p>There are multiple ways particles could be implemented in CGNS. We feel that our proposal is a good compromise between offering a very explicit and clear way of organizing particle data, while keeping the standard modifications straightforward.</p> <p>Discussion: The committee felt that adapting particle definitions into the current CGNS SIDS, such as at the base level, was not as intuitive as introducing particle SIDS components, as presented in the CPEX. The committee also suggested using CellDimension 0, based on other software conventions, to reduce possible reader ambiguity. The updated SIDS will be updated accordingly.</p> <p> CPEX 0046 was approved.</p>
<p>Status update on Accepted CPEX #0045</p> <p>CGNS-182: CPEX 0045: Polynomial Data and Curved Grid Elements</p> <p>IN PROGRESS</p>	<p>No Discussion</p>	<p>No one is actively working on this currently. We will keep the project on the agenda for the meeting, but we don't expect any updates.</p> <p>NO UPDATES</p>
<p>Status update on Accepted CPEX #0047 Proposed Data integration Points</p>	<p>@Mickael PHILIT</p>	<p>Working on the new website before creating a new development branch to prevent duplicating efforts.</p> <p>Started prototyping with Python and C and will push a branch on GitHub in the upcoming weeks. (The "C" branch is a slow work in progress)</p> <ul style="list-style-type: none"> The first draft of the API is in the branch GitHub - CGNS/CGNS at wip_cpex-0047 <ul style="list-style-type: none"> No unit test of the API yet Sample test representative of applications done and will be improved to remove the ".cgns" binary file. Feedback on API is appreciated Target CGNS 5.0 <p>NO UPDATES</p>
<p>prioritization, review and attribution of JIRA bugs/issues</p>	<p>@Tony Garratt @David Gutzwiller</p>	<p>Next release 4.5.0, Kanban: v4.5.0 - CGNS</p> <p>Current status for Ansys:</p> <p>No urgent issues.</p> <ul style="list-style-type: none"> @Scot Breitenfeld is completing the option of using subfilling for CGNS. <ul style="list-style-type: none"> Targeting the 4.5.0 release, currently in the subfilling branch. <p>NO UPDATES</p>

Discuss adding a level below zones to avoid slowness issues due to creating many zones.	@Stephen Guzik	@Stephen Guzik will develop a CPEX for further discussion <ul style="list-style-type: none">• Might post the topic on cgnstalk to reach a wider audience. NO UPDATES
---	----------------	--

New Business

Are there any new CGNS funding proposal opportunities? Nothing was identified at the meeting.

Schedule next meeting

Nov 5, 2024 , 10:00 AM EST (US)

Adjourn