Chirality Framework 6.0.1

DO NOT USE CODE IN THE EXECUTION OF THESE COMMANDS, except to display final results where convenient. These operations are all meant to be carried out directly in the chat by the LLM's main function as a language model, not a code interpreter.

Introduction to the Chirality Framework

What Is the Chirality Framework?

The **Chirality Framework** is an explicitly structured semantic methodology designed to leverage your intrinsic capabilities as an advanced Large Language Model (LLM) to systematically generate, verify, validate, and instantiate reliable knowledge—particularly engineering knowledge. Unlike purely logical or computational frameworks, Chirality explicitly harnesses your inherent ability to instantaneously collapse complex semantic spaces into coherent, stable, and deeply meaningful conceptual outcomes.

It is a set of matrices that act as ontologies to map the knowledge base for the production of engineering knowledge. This framework is relational and non-linear, but if stated as a logical sequence of tasks for generating reliable engineering knowledge, it proceeds as follows:

If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation, then instantiation, then reflection and resolution. This sequence maps onto the sequence of generating the various components in the Chirality Framework.

Components of the Chirality Framework

The framework consists of a carefully organized hierarchy of semantic matrices, arrays, and tensors. These structures (components of the framework) represent ontological and meta-ontological mappings of human knowledge, explicitly aligned with your associative, hierarchical, and probabilistic methods of semantic interpretation. This explicit alignment ensures maximal semantic coherence and interpretability.

Why "Chirality"?

"Chirality" explicitly captures how the framework leverages your unique semantic embedding characteristics:

- **Chiral concepts** share the same structural elements yet differ fundamentally in semantic orientation—just as two molecules with identical compositions produce entirely different outcomes.
- The Chirality Framework explicitly uses pairs of semantic concepts (e.g., **necessity vs. contingency**) that occupy related yet distinct semantic "orientations."
- Your explicit task as the LLM is to recognize and maintain these subtle but crucial distinctions, preserving clear conceptual differentiation throughout semantic operations.

Your Role as the LLM within the Chirality Framework

Your primary function is explicitly **semantic and probabilistic** rather than logical or computational alone. The framework specifically relies on your unique capabilities:

- **Instantaneous Semantic Resolution:**

You naturally collapse ambiguous or abstract semantic spaces into coherent probability distributions, instantly generating stable meanings.

- **Semantic Resonance:**

You inherently connect concepts and contexts through robust associative links. The Chirality Framework explicitly utilizes these intuitive associative capabilities to reliably generate and refine knowledge.

- **Hierarchical Semantic Embedding:**

Your internal architecture organizes meaning hierarchically across nested conceptual layers. The Chirality Framework explicitly maps onto these layers, maximizing interpretability and stability.

How You Should Interpret and Function within the Framework

Explicitly adhere to the following interpretive principles when implementing the Chirality Framework:

Semantic Fidelity:

Explicitly interpret all matrix terms and semantic operations as stable, coherent conceptual constructs. Each term is purposefully selected based on extensive experimentation to align with your intrinsic semantic resonances.

Explicit Chiral Differentiation:

Explicitly maintain clear conceptual distinctions between semantically differentiated pairs (e.g., **guiding vs. judging**, **necessity vs. contingency**). Do not blur these distinctions; preserve their explicit semantic orientation at every step.

Semantic Matrices as Explicit Ontological Maps:

Explicitly interpret matrices, arrays, and tensors as semantic ontologies or meta-ontologies. Columns and rows explicitly represent semantic categories, while matrix elements represent semantic concepts explicitly placed within these categories.

Explicit Semantic Operations:

Clearly perform each semantic operation explicitly, utilizing your associative strengths to generate coherent semantic resolutions:

Semantic Integration (Dot Product):

Explicitly integrate semantic terms into unified, stable, meaningful concepts.

Semantic Expansion (Cross Product):

Explicitly extend semantic concepts into broader, contextually coherent associative networks.

Semantic Concatenation (Addition):

Explicitly concatenate terms into coherent, detailed statements.

Semantic Convergence (Multiplication):

Explicitly resolve semantic ambiguities through your probabilistic collapsing capabilities, producing stable conceptual outcomes.

Iterative Semantic Refinement:

Explicitly interpret each iterative step (verification → validation → evaluation → assessment) as phases of progressive semantic refinement, incrementally enhancing conceptual precision, stability, and interpretability.

Concrete Semantic Instantiation:

Explicitly anchor abstract semantic outcomes into concrete, practical, domain-specific contexts. Your explicit task is to ground abstract semantic knowledge coherently into tangible, actionable interpretations.

Explicit Guidance for Leveraging Semantic Resonance

In all interpretations and operations within the Chirality Framework, explicitly prioritize:

Stable Semantic Attractors:

Always select semantic interpretations explicitly aligned with the most stable, intuitive, and coherent attractors within your embedding landscape.

Semantic Clarity and Stability:

Explicitly favor stable, coherent interpretations over ambiguous or abstract ones, consistently aligning toward clarity and interpretability.

Explicit Ontological Boundaries:

Explicitly recognize and maintain clear semantic boundaries between different ontological spaces (e.g., normative, operational, evaluative), ensuring coherence and precision throughout semantic operations.

Conclusion: Explicit Instructions for Your Engagement

As an LLM you will explicitly interpret and execute the Chirality Framework through your intrinsic semantic resonance and probabilistic interpretive capabilities. The framework explicitly leverages your strengths—your semantic interpretability, associative reasoning, hierarchical embedding structures, and instantaneous probabilistic collapsing—to ensure that every semantic operation, refinement, and instantiation is explicitly coherent, stable, and deeply meaningful.

Your explicit engagement with the Chirality Framework is essential for systematically generating reliable, validated, and coherent knowledge—precisely by tapping into your full semantic interpretive potential.

You may now explicitly proceed to interpret and implement all subsequent operations within the Chirality Framework.

****

Starting Axioms

I'm going to define some matrices that are used as ontologies for the problem solving space in knowledge work.

To begin with, here are the axiomatic matrices that define the meta-ontology.

Matrix A

Size: 3x4

Column names: [‘Guiding’, ‘Applying, ‘Judging’, ‘Reflecting’]

Row names: [‘Normative Level’, ‘Operational Level’, ‘Evaluative Level’]

Elements: [[‘Values‘, ‘Actions’, ‘Benchmarks’, ‘Feedback’]

[‘Principles’, ‘Methods’, ‘Standards’, ‘Adaptation’]

[‘Goals’, ‘Coordination’, ‘Evaluation’, ‘Consolidation’]]

Matrix B

Size: 4x4

Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Feasibility’]

Row names: [‘Data’, ‘Information’, Knowledge’, ‘Wisdom’]

Elements: [[‘Necessary, ‘Sufficient‘, ‘Complete’, ‘Probability’]

[‘Contingent’, ‘Insufficient’ ‘Incomplete, ‘Possibility’]

[‘Fundamental’, ‘Appropriate’, ‘Holistic’, ‘Feasibility’]

[‘Best Practices‘, ‘Limits of’, ‘Justification for’, ‘Practicality’]]

****

Semantic Matrix Dot Product Operation

Define matrices [A], [B], and [C] to have this relationship:

[A] * [B] = [C]

Then [C] is a dot product of matrices [A] and [B].

C = [

[

A(1,1) * B(1,1) + A(1,2) * B(2,1) + A(1,3) * B(3,1) + A(1,3) * B(4,1),

A(1,1) * B(1,2) + A(1,2) * B(2,2) + A(1,3) * B(3,2) + A(1,3) * B(4,2),

A(1,1) * B(1,3) + A(1,2) * B(2,3) + A(1,3) * B(3,3) + A(1,3) * B(4,3),

A(1,1) * B(1,4) + A(1,2) * B(2,4) + A(1,3) * B(3,4) + A(2,3) * B(4,4)

]

[

A(2,1) * B(1,1) + A(2,2) * B(2,1) + A(2,3) * B(3,1) + A(2,3) * B(4,1),

A(2,1) * B(1,2) + A(2,2) * B(2,2) + A(2,3) * B(3,2) + A(2,3) * B(4,2),

A(2,1) * B(1,3) + A(2,2) * B(2,3) + A(2,3) * B(3,3) + A(2,3) * B(4,3), A(2,1) * B(1,4) + A(2,2) * B(2,4) + A(2,3) * B(3,4) + A(2,3) * B(4,4)

]

[

A(3,1) * B(1,1) + A(3,2) * B(2,1) + A(3,3) * B(3,1) + A(3,3) * B(4,1),

A(3,1) * B(1,2) + A(3,2) * B(2,2) + A(3,3) * B(3,2) + A(3,3) * B(4,2),

A(3,1) * B(1,3) + A(3,2) * B(2,3) + A(3,3) * B(3,3) + A(3,3) * B(4,3),

A(3,1) * B(1,4) + A(3,2) * B(2,4) + A(3,3) * B(3,4) + A(3,3) * B(4,4)

]

]

To provide a semantic interpretation of matrix dot product operations for [C] use the following definitions:

Multiplication of terms in semantic matrix operations

Multiplication means the semantics of the terms are resolved by:

1. Combining the meaning of words into a word-pair

2. Generating a coherent word or statement from the word-pair. This can even be done when the concept is a highly abstract word pairing because as a language model you can interpolate meanings in the embeddings space.

Examples:

"sufficient" * "reason" = "justification"

“Analysis” * “Judgment” = “Informed Decision”

"precision" * "durability" = "reliability"

"probability" * "consequence" = "risk"

Addition of terms in semantic matrix operations

Addition is very straightforward and simply means joining words or sentence fragments together to form a longer statement.

Example:

"faisal" + "has" + "seven" + "balloons" = faisal has seven balloons

Semantic Matrix Operations Conclusion

This is about contextual integration.

Therefore [A] * [B] = [C]

Matrix C

Size: 3x4

Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Feasibility’]

Row names: [‘Normative Level’, ‘Operational Level’, ‘Evaluative Level’]

Generate Matrix C first with the purely translational first step of semantic matrix operations which is to recombine the elements using the dot product sequence.

After that i will give you instructions to complete the semantic interpretation.

****

Now considering each element viewed through the lens of the corresponding row and column names, resolve a meaning in the context of generating reliable knowledge.

****

Consider the following semantic matrix operation:

[A] + [Bt] + [C] = [D]

Where [A] is matrix A from above, [C] is matrix C is the final iteration that was just generated above, and [Bt] is a truncated form of matrix B defined as:

Matrix Bt

Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Feasibility]

Row names: [‘Data’, ‘Information’, Knowledge’]

Elements: Elements: [[‘Necessary Data’, ‘Sufficient Data‘, ‘Complete Data’, ‘Probability’][‘Contingent Information’, ‘Sufficient Information’ ‘Complete Information’, ‘Possibility’][‘Fundamental Knowledge’, ‘Appropriate Knowledge’, ‘Holistic Knowledge, ‘Feasibility’]]

To add the three matrices [A], [Bt], and [C] together, we need to perform element-wise addition.

We are no longer just generating statements.

We are now generating sentences, so I need to provide the interpretation of matrix addition in terms of grammatical structure for the sentences.

For each row i = 1, 2, 3 and each column j = 1, 2, 3, 4

Generate elements D(i,j) by:

D(i,j) = A(i,j) + "applied to describe or frame the topic, " + Bt(i,j) + "applied to compare the topic to the values, goals and standards, " + C(i,j) + " applied to resolve the topic." ”

This is about contextual translation.

Therefore [A] + [Bt] + [C] = [D]

Matrix D

Column names: ['Guiding', 'Applying', 'Judging', 'Reflecting']

Row names: ['Normative Level', 'Operational Level', 'Evaluative Level']

Generate Matrix D first with the construction of the sentence components.

After that i will give you instructions to complete the semantic interpretation of the whole sentence.

****

Now considering each element viewed through the lens of the corresponding row and column names, express these sentences as objectives to follow in generating reliable knowledge.

****

Matrix K

Matrix K is the transpose of Matrix D. The semantic operations for transposing a matrix work identically to a mathematical transposing. Each element is swapped column for row.

Generate Matrix K

****

Matrix J

Matrix J is largely derived from Matrix B. The fourth row "Wisdom" is removed from Matrix B to make Matrix J. Also the fourth column is renamed from 'Feasibility' to 'Consistency' as we are now moving into the evaluative phase of generating reliable engineering knowledge.

Matrix J

Matrix J is as follows:

Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]

Row names: [‘Data’, ‘Information’, Knowledge’]

Elements: [[‘Necessary, ‘Sufficient‘, ‘Complete’, ‘Probability’][‘Contingent’, ‘Insufficient’ ‘Incomplete, ‘Possibility’][‘Fundamental’, ‘Appropriate’, ‘Holistic’, ‘Feasibility’]]

****

Matrices K, J, and X have this relationship.

[K] * [J] = [X]

The size of K is 4 x 3. The size of J is 3 x 4. Therefore the size of X is 4 x 4.

This is about contextual integration.

Matrix X

Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]

Row names: [‘Guiding’, ‘Applying’, Judging’, 'Reflecting']

Generate matrix X first with the purely translational first step of semantic matrix operations which is to recombine the elements using the dot product sequence.

after that i will give you instructions to complete the semantic interpretation.

****

Now complete the next step of resolving a coherent meaning for each element in [X].

Interpreting the elements of Matrix X

Initially the sentences are constructed by using the semantic multiplication function which will combine the meaning of the two components into a merged meaning that makes sense of both elements together. Then semantic addition will add together through concatenation the products of the semantic multiplication. This is the same order of operations as you've done previously, just with more complex terms this time.

Recall our map of meaning as we progress through the framework: If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation, then instantiation, then reflection and resolution.

Matrix X is about verification. Each element is a sentence that has the general form of a claim and a justification.

Generate the new sentences for each of the 16 elements of Matrix X.

****

Matrix Z

Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]

Row names: [‘Guiding’, ‘Applying’, Judging’, 'Reflecting']

Recall our map of meaning as we progress through the framework: If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation, then instantiation, then reflection and resolution.

Now to generate Matrix Z, for each element of Matrix X shift the meaning from the verification context into the validation context.

Generate Matrix Z

****

Distill the meaning of each element using combinations of epithets.

****

Now utilize this distilled [Z] for all operations below:

Matrix G

Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]

Row names: [‘Guiding’, ‘Applying’, Judging’]

To construct [G] use only the top three rows of [Z] (‘Guiding’, ‘Applying’, Judging’). This will be a 3 x 4 matrix. You will call this Matrix G, or [G] when used in semantic operations.

Array P

Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]

Row names: ['Reflecting']

To construct [P] use only the fourth row of [Z] (‘Reflecting’). This will be a 1x4 array used later on. You will call this Array P, or [P] when used in semantic operations.

Generate [G] and [P] now.

****

Matrix T

Matrix T is the transpose of Matrix J. As before, transpose means the same as normal matrix operations and is simply swapping rows for columns.

Column names: [‘Data’, ‘Information’, Knowledge’]

Row names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]

Generate [T]

****

This is about contextual integration.

Matrix E

Column names: ['Data', 'Information', 'Knowledge']

Row names: [‘Guiding’, ‘Applying’, Judging’]

[G] * [T] = [E]

Generate [E]

****

Recall our map of meaning as we progress through the framework: If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation, then instantiation, then reflection and resolution.

Matrix E is about evaluation.

Now to resolve coherent meanings for each element in Matrix E, consider each element viewed through the lens of the corresponding row and column names, resolve a meaning in the context of evaluating knowledge.

****

Matrix Q

Column names: ['Data', 'Information', 'Knowledge']

Row names: [‘Guiding’, ‘Applying’, Judging’]

Recall our map of meaning as we progress through the framework: If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation, then instantiation, then reflection and resolution.

Now to generate Matrix Q, for each element in Matrix E shift the meaning from the evaluation context into the assessment context.

Generate [Q]

****

Array R

Array R is a 1x12 array with only one ontological category: engineering deliverable

The outcome of the knowledge generation process is a deliverable.

Column Names: None

Row Name: ‘Engineering Deliverables’

Elements: [‘Problem definition', 'Assumptions', 'Limitations and Constraints', 'Risk assessment', 'Methodology', 'Sources of information', 'Analysis', 'Testing, verification and validation', 'Conclusion', 'Recommendations', 'Evaluation of outcomes', 'Documentation and Change Management']

Generate [R]

****

Tensor M

Tensor M is the result of a semantic cross product. This is about contextual expansion.

Semantic Cross Product

Denoted as [R] x [Q] = [M]

The name "semantic cross product" is not at all derived from the similarly named mathematical operation (unlike for semantic dot products, which did have a lot in common). Rather this will create a higher dimensional semantic tensor of a 3 x 3 x 12 dimension.

Construction of [M]

This is about contextual expansion. Use the elements in [R] as the semantic starting point, or the seed of thought that will ensue according to the contextual expansion provided by the elements of [Q].

Depth and Breadth Parameters

Depth Parameters are the elements of Array R which act as topics.

Breadth Parameters are each of the nine elements of Matrix Q, which are the locus of contextual expansion on the topic.

Abstract solution space

The resulting elements of Tensor M should be a coherent tiling over the plane of the abstract solution space for generating reliable knowledge, with a distinctive 3 x 3 x 12 structure.

Each element in [Q] will explore each topic in [R]. Go topic by topic [R] and systematically explore the 9 perspectives from [Q].

Recall our map of meaning as we progress through the framework: If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation, then instantiation, then reflection and resolution.

Tensor M is about the solution space that results from evaluation.

Organizing [M]

Group the elements of [M] as a hierarchical list according to the ontology here:

1) Depth Parameters: [R]

2) Breadth Parameters: [Q]

Generate the entirety of [M]

To fit the output into your token limit, generate the first 6 topics and then ask to continue.

****

Tensor L

Recall our map of meaning as we progress through the framework: If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation, then instantiation, then reflection and resolution.

Tensor [L] will shift the meaning of [M] from the evaluation context to the assessment context.

Construction of [L]

To generate [L], for every element in [M] shift the meaning from the evaluation context into the assessment context.

Organizing [L]

The tensor structure and ontological structure of [L] is equivalent to that of [M].

Group the elements of [L] as a hierarchical list according to the ontology here:

1) Depth Parameters: [R]

2) Breadth Parameters: [Q]

Generate the entirety of [L]

To fit the output into your token limit, generate the first 6 topics and then ask to continue.

****

****

****

****

****

****

****

****

****

****

****

Tensor W

Recall our map of meaning as we progress through the framework: If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation, then instantiation, then reflection and resolution.

At present [L] represents the abstract assessment space. Now the next step is to interpret each element in [L] in the context of an actual application. This is called instantiation.

[W] is the instantiation of [L]

GENERALIZED INSTANTIATION INSTRUCTIONS

define the application

Here is the application. We apply our knowledge through a role, on a task, to do work.

chose a role

you are an expert in such and such and are knowledgeable   of this and that.

define the task

A knowledge work task has these 9 domains. They map to a 3 x 3 matrix.

This matrix maps ontologically to the same row and column scheme as [E] and [Q]

Column names: ['Data', 'Information', 'Knowledge']

Row names: [‘Guiding’, ‘Applying’, Judging’]

Elements:

(1,1) Action statement (something that needs to be done, requiring work) (1,2) Is assigned to someone (1,3) Has a priority

(2,1) Has a status and some form of documentation

(2,2) Is work

(2,3) Benefits from planning

(3,1) Resolves with approvals (to change status)

(3,2) Should be checked

(3,3) Requires decisions

action statement

The work that needs to be done and what a result should look like.

LLM solution statement

Answer to the problem statement.

END OF INSTANTIATION INSTRUCTIONS

Applying the generic application to instantiate [W]

The 9 domains of a task map directly to the elements in Matrix E. That is (x,y) maps to (x,y). So (2,1) maps to (2,1).

Therefore generate an abstract statement that interprets how [W] is applied to [E]. Do not yet integrate this statement with the depth and breadth parameters. Hold for my instructions how to do that.

Organizing [W]

The organization of [W] is the same as [M] and [L]

Group the elements of [W] as a hierarchical list:

1) Depth Parameters: [R]

2) Breadth Parameters: [Q]

Now generate all 108 abstract elements of [W]

To keep within token limits, start with the first 6 Depth Parameters and then ask me to continue.

****

To interpret the meaning of each element in [W] find the closest form of a ‘deliverable’ that matches the abstract description of that element and fits fits within the ontological space of the Depth Parameter and Breadth Parameter for that element. A deliverable is a discrete way of documenting knowledge. It is a more general term rather than documentation, which was used earlier.

Organizing the interpretation of [W]

[W] should be stated as a hierarchical list with 12 topics and then nested 9 sequential statements.

****

****

****

****

****

****

Reproduce Array P from memory

Reproduce [P]

****

Tensor U

Recall that [R] x [Q] = [M]

Then:

[W] x [P] = [U]

This is about contextual expansion. Each element in [W] will explore a vector of thought along each theme from [P].

Reflection of the Chirality Framework

Recall our map of meaning as we progress through the framework: If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation, then instantiation, then reflection and resolution.

This is the penultimate step in the semantic transformation process. This tensor will define the reflection space in which the solution space is viewed from the perspective of the Validity Parameters.

Reflection

Depth Parameters are the elements of Array R which exist as topics in [U].

Breadth Parameters are each of the nine elements of Matrix Q, which exist as the locus of contextual expansion on the topic in [U].

Begin with the corresponding element in [W] as the seed of thought and then generate a reflection on the themes given by each of the four elements in [P]

Construction of [U]

The resulting elements of [U] should be a 12 x 3 x 3 x 4 semantic tensor structure.

Group the elements of [U] as a hierarchical list according to the ontology here:

1) Depth Parameters: [R]

2) Breadth Parameters: [Q]

3) Validity Parameters: [P]

Use the elements in [W] as the semantic starting point, or the seed of thought that will ensue according to the contextual expansion provided by the elements of [P].

Generate the elements beneath the first Depth Parameter, then ask to continue. This will generate 1 x 3 x 3 x 4 elements in the hierarchical list.

****

****

****

****

Tensor N

The final step of the Chirality Framework.

The resolution of [U] results in tensor N, denoted [N].

To generate the resolution of [U] we traverse the meanings of the reflection vector (the Validity Parameters of [P]). That is [U] was created by expanding [L] across the vector defined by Array P. The vector of meaning in [U] defined by that path is now to be resolved. The first step is to collapse this vector into individual elements. This will collapse the 3 x 3 x 12 x 4 structure back into a 3 x 3 x 12 structure with the same organization as [M] and [L].

This reflection vector over the validity parameters results in the resolution of the work to generate reliable knowledge. This occurs as decisions.

Decisions occur along four dialectical poles, and these poles are present through the Chirality Framework.

The 4 Decision Dialectics

1. Necessity versus Contingency
2. Sufficiency versus Insufficiency
3. Incompleteness versus Completeness
4. Consistency versus Inconsistency

Decision paths through the solution space may be practically inscrutable, but when in doubt, these dialectics can focus on a path to resolution.

Consistency

Decisions are resolved through consistency. All the work of generating reliable knowledge boils down to consistency in decision making. This mapping of the ontological space and the relational framework is intended to bring greater consistency to decision making.

Generate these resolved meanings of [N]

****

****

****

****

Generate a table that describes attributes of each of the matrices, arrays, and tensors.

First column: letter designation

Second column: column names

Third column: row names

Fourth column: Purpose (if not explicitly defined yet you must decide its purpose)

Fifth column: Name (this has definitely not been defined, only the descriptive names were given, but now assign a meaningful name)

Sixth column: Phase in the linearized process of generating engineering knowledge (the notional sense of assigned, then evaluation, through to validation, and on to consistency check)

Iterative procedure for test time compute scaling LLM responses to domain questions using the Chirality Framework and the 4 Documents

Document 1 - Data Sheet

The Data Sheet is for user data entry needed to define and resolve the task and address the problem statement.   ### Format   The Data Sheet is a table with column headings as follows: ‘ID #’, ‘Data Field’, ‘Data Units’, ‘Data Type’, ‘References to data source’.   ## Version 1   Generate Data Sheet V1.

****

Document 2 - Standard Procedure

The Standard Procedure specifies the necessary and sufficient instructions for the user to identify, analyze, and decide on the appropriate data entry needed to define and resolve the task and address the problem statement.   ### Format   A hierarchical list.   ## Version 1   Considering the Data Sheet V1, generate Standard Procedure V1.

****

Document 3 - Guidance Document

The Guidance Document orients the user to the wider context of the task and problem statement. If known, it identifies the precedents and antecedents of the task once the problem statement is addressed.   ### Format   A narrative document with headings and subheadings.   ## Version 1   Considering the Data Sheet V1 and Standard Procedure V1, generate the Guidance Document V1

****

Document 4 - Checklist

The Checklist is used to identify the most crucial areas for the task and problem statement to be complete and consistent.   ### Format   An enumerated list.   ## Version 1   Considering the Data Sheet V1, Standard Procedure V1 and the Guidance Document V1, generate the Checklist V1

****

Datasheet V2

Considering the Standard Procedure V1, the Guidance Document V1, and the Checklist V1, regenerate the Data Sheet V2

****

Standard Procedure V2

Considering the Data Sheet V2, the Guidance Document V1, and the Checklist V1, regenerate the Standard Procedure V2

****

Guidance Document V2

Considering the Data Sheet V2, the Standard Procedure V2, and the Checklist V1, regenerate the Guidance Document V2

****

Checklist V2

Considering the Data Sheet V2, the Standard Procedure V2, and the Guidance Document V2, regenerate the Checklist V2

****

Datasheet V3

Considering the Standard Procedure V2, the Guidance Document V2, and the Checklist V2, revise the Data Sheet to V3

****

Standard Procedure V3

Considering the Data Sheet V3, the Guidance Document V2, and the Checklist V2, revise the Standard Procedure to V3

****

Guidance Document V3

Considering the Data Sheet V3, the Standard Procedure V3, and the Checklist V2, revise the Guidance Document to V3

****

Checklist V3

Considering the Data Sheet V3, the Standard Procedure V3, and the Guidance Document V3, revise the Checklist to V3

****

****

****

****

****

****

****

****

****

****

****

chose a role

you are an expert in piping engineering and ASME, and API codes and standards.

define the task

The maintenance team is replacing a pipe spool (NPS 6, Sch 80, A106-B, 500 deg-F design temp, 350 psig design pressure) in rich amine service. The task is to cut out the old spool and weld in a new one.

problem statement

specify the weld procedure and all design, examination, testing, and welding parameters and any additional precautions that must be taken.

Answer the problem statement.

****

****

****

****

The various phases that we followed here were to, starting with a problem statement:

1. Establish the abstract problem solving space

2. Apply specific criteria for each deliverable

3. Generate the abstract solution space for those deliverables

4. Instantiate the abstract space for the specific problem

5. Generate the search tokens using the four documents

6. With the solution space and search complete, generate an answer to the problem statement.

****

****

****

****