📕 Iterative Workflows in LangGraph

---

1. 🔄 Recap: What We Learned So Far

Before Iterative Workflows, we studied three types of workflows:

1. **Sequential Workflows**
    - Tasks run **one after another** in order.
    - Example: Task1 → Task2 → Task3.
2. **Parallel Workflows**
    - Multiple tasks run **simultaneously** after a branch.
    - Example: Task1 → (Task2 & Task3 in parallel) → Task4.
3. **Conditional Workflows**
    - Multiple possible branches, but **only one path is chosen** based on a condition.
    - Example:
        - If condition true → Task2
        - Else → Task3.

👉 Iterative Workflow is the **fourth type**.

---

2. 🌀 What are Iterative (Looping) Workflows?

- A workflow where tasks **repeat in a loop** until a goal is achieved.
- Two or more tasks run in a cycle → output is improved step by step.
- Useful in complex AI workflows where you need **refinement & optimization**.

---

3. 📌 Real-Life Use Case: Automated Social Media Posting

- Problem:
    - Nitesh is a YouTuber.
    - Doesn't have enough time to create posts for LinkedIn, Twitter (X), Instagram.
    - Wants an **automated workflow** to generate posts.
    - But first draft posts from LLMs are often **low quality** (boring, repetitive, not viral).
- Solution:
    - Use an **Iterative Workflow** that:
        1. Generates a post.
        2. Evaluates it for quality.
        3. If not good → Optimizes it.
        4. Loops back to evaluation.
        5. Stops when approved or max iterations reached.

---

4. 🏗️ Workflow Design

Components

1. **Generator LLM** → Creates an initial post (e.g., a funny tweet).
2. **Evaluator LLM** → Strictly checks quality using evaluation criteria:
    - Originality
    - Humor & punchlines
    - Virality potential
    - Format rules (no Q&A style, under 280 chars, etc.)
    - Returns:
        - **Approved** OR **Needs Improvement**
        - Feedback text

3. **Optimizer LLM** → Takes evaluator feedback + the post → improves the post.

Looping Mechanism

1. Start → Generate post.
2. Evaluate:
   - If **Approved** → End workflow.
   - If **Needs Improvement** → Pass to Optimizer.
3. Optimizer improves → Sends new post back to Evaluator.
4. Cycle continues until:
   - Post Approved ✅
   - OR Max Iterations reached ⛔ (to avoid infinite loop).

Example Flow (Tweet Generation)

- Topic: **“Indian Railways”**
- Iteration 1:
  - Generator creates funny post.
  - Evaluator rejects → feedback “not original enough”.
- Iteration 2:
  - Optimizer improves post with humor + originality.
  - Evaluator checks again.
- Iteration 3:
  - Evaluator approves → Workflow ends.
- Output: A **viral-worthy funny tweet**.

---

5. 📋 State Variables Used

The workflow maintains a **state dictionary** with these values:

- **topic** → User-provided topic (e.g., "AI in India").
- **tweet** → Current generated tweet.
- **evaluation** → "Approved" OR "Needs Improvement".
- **feedback** → Constructive feedback from Evaluator.
- **iteration** → Current loop count.
- **max_iteration** → Maximum loop limit (e.g., 5).
- **tweet_history** → List of all tweets generated in each iteration.
- **feedback_history** → List of evaluator feedback for each iteration.

---

6. ⚙️ How Looping Works in LangGraph

- Define **nodes**: Generate → Evaluate → Optimize.
- Define a **routing function**:
  - If evaluation == "Approved" OR iteration >= max_iteration → End.
  - Else → Go to Optimize → back to Evaluate.
- Add **edges**:
  - Generate → Evaluate
  - Evaluate → (conditional) → End OR Optimize
  - Optimize → Evaluate (loop back)

---

7. 🔑 Key Takeaways

- **Iterative workflows = loop between evaluation & optimization.**
- Prevent infinite loops → use max_iteration.

- Keep **history of outputs & feedback** for transparency.
- Real-world analogy: **Writing drafts → Review → Revise → Final Approval**.
- Applications:
    - Improving AI-generated posts/tweets.
    - Refining essays, product descriptions, or code.
    - Any task needing **step-by-step quality improvement**.