To store lines from a file into a std::vector<std::vector<std::string>>, you need to read the file line by line, split each line into individual "words" or "tokens," and then store those tokens as a std::vector<std::string> inside the main std::vector<std::vector<std::string>>.
Here's a step-by-step approach with a complete C++ example.

Step 1: Include Necessary Headers

You'll need <iostream>, <fstream>, <string>, <vector>, and <sstream> for file handling, strings, vectors, and string streams, respectively.

Step 2: Open the File and Check for Errors

Always check if the file opened successfully. If not, print an error message and exit.

Step 3: Loop Through the File Line by Line

Use a while loop with std::getline() to read each line from the file until the end.

Step 4: Tokenize Each Line

For each line, create a std::stringstream to treat the line as an input stream. Then, use another while loop to extract each "word" (delimited by spaces) and store it in a temporary std::vector<std::string>.

Step 5: Store the Tokenized Line

After tokenizing a line, push the temporary vector of strings into your main std::vector<std::vector<std::string>>.

Here is the complete code example:

C++

```
#include <iostream>
#include <fstream>
```

```
#include <string>
#include <vector>
#include <sstream>

int main() {
    // A. Define the data structure
    std::vector<std::vector<std::string>> data;

    // B. Open the file
    std::ifstream inputFile("my_file.txt");
    if (!inputFile) {
        std::cerr << "Error: Could not open the file." << std::endl;
        return 1;
    }

    // C. Read the file line by line
    std::string line;
    while (std::getline(inputFile, line)) {
        // Create a temporary vector for the words in the current line
        std::vector<std::string> words;

        // Use stringstream to split the line into words
        std::stringstream ss(line);
        std::string word;
        while (ss >> word) {
            words.push_back(word);
        }

        // Add the vector of words to the main data structure
        data.push_back(words);
    }

    // D. Close the file
    inputFile.close();

    // E. (Optional) Display the contents to verify
    std::cout << "Contents of the vector<vector<string>>:" << std::endl;
    for (const auto& inner_vec : data) {
        for (const auto& w : inner_vec) {
            std::cout << w << " ";
        }
```

```
        std::cout << std::endl;
    }

    return 0;
}
```

Assuming my_file.txt contains:

```
Hello world
This is C++
Programming is fun
```

The program will produce the output:

```
Contents of the vector<vector<string>>:
Hello world
This is C++
Programming is fun
```

Explanation of Key Components

- **std::vector<std::vector<std::string>> data;**: This is the main data structure where all the file's content will be stored. It's a vector of vectors, where each inner vector represents a line and contains strings (the words of that line).
- **std::stringstream ss(line);**: This is a powerful class from the <sstream> header. It allows you to treat a string (line) as an input stream, just like std::cin or a file stream. This makes it easy to extract words using the >> operator.
- **while (ss >> word)**: This loop reads "words" from the stringstream, with spaces acting as the default delimiter. The loop continues until there are no more words to extract. Each word is stored in the word string variable.
- **words.push_back(word);**: Each extracted word is added to the temporary vector words.
- **data.push_back(words);**: After the inner loop finishes processing a full line, the temporary words vector is pushed into the main data vector, storing the entire line's content.