```
#include <iostream>
#include <vector>
#include <cmath>

using namespace std;

// Hz 到 Mel 的转换
double hz_to_mel(double hz) {
    return 2595.0 * log10(1.0 + hz / 700.0);
}

// Mel 到 Hz 的转换
double mel_to_hz(double mel) {
    return 700.0 * (pow(10.0, mel / 2595.0) - 1.0);
}

// 生成 Mel 滤波器组
vector<vector<double>> mel_filter_bank(int num_filters, int fft_size, int sample_rate) {
    vector<vector<double>> filters(num_filters, vector<double>(fft_size / 2 + 1, 0.0));

    // 计算 Mel 频率点
    double min_mel = hz_to_mel(0.0);
    double max_mel = hz_to_mel(sample_rate / 2.0);
    vector<double> mel_points(num_filters + 2);
    vector<double> hz_points(num_filters + 2);
    vector<int> bin_points(num_filters + 2);

    for (int i = 0; i < num_filters + 2; i++) {
        mel_points[i] = min_mel + (max_mel - min_mel) * i / (num_filters + 1);
        hz_points[i] = mel_to_hz(mel_points[i]);
        bin_points[i] = static_cast<int>(hz_points[i] * fft_size / sample_rate);
    }

    // 构造滤波器
    for (int i = 1; i <= num_filters; i++) {
        for (int j = bin_points[i - 1]; j < bin_points[i]; j++) {
            filters[i - 1][j] = (j - bin_points[i - 1]) / static_cast<double>(bin_points[i] -
bin_points[i - 1]);
        }
        for (int j = bin_points[i]; j < bin_points[i + 1]; j++) {
            filters[i - 1][j] = (bin_points[i + 1] - j) / static_cast<double>(bin_points[i + 1] -
bin_points[i]);
        }
    }
```

```
    return filters;
}

int main() {
    int sample_rate = 16000;   // 采样率
    int fft_size = 512;        // FFT 点数
    int num_filters = 26;      // 滤波器数量

    // 生成 Mel 滤波器组
    vector<vector<double>> filters = mel_filter_bank(num_filters, fft_size, sample_rate);

    // 打印第一个滤波器
    cout << "Mel Filter 0:" << endl;
    for (int i = 0; i < filters[0].size(); i++) {
        cout << i << ": " << filters[0][i] << endl;
    }

    return 0;
}
```