1 Execution Example

Here is the example of a dynamic circuit discussed in section 2.2.1, written in OpenQASM2.0:

```
OPENQASM 2.0;

gate u2(phi,lambda) q { U(pi/2,phi,lambda) q; }
gate h a { u2(0,pi) a; }

qreg q0[1];
creg c0[1];
creg c1[1];

h q0[0];
measure q0[0] -> c0[0];
if(c0==0) h q0[0];
measure q0[0] -> c1[0];
```

Note that the values in the classical registers, are either 00, 10, or 11, with probabilities of 1/4, 1/4, and 1/2, respectively.

```
$ dune exec qasm example/basic.qasm
QASMCore ========================================
RotateInstr (1.570796, 0.000000, 3.141593, 0)
MeasureInstr (0, 0)
IfInstr (0, false,
RotateInstr (1.570796, 0.000000, 3.141593, 0)
NopInstr)
MeasureInstr (0, 1)
NopInstr
RESULT ==========================================
00 : 2.5000000000000011e-01
01 : 0.0000000000000000e+00
10 : 2.5000000000000000e-01
11 : 4.9999999999999989e-01
```

2 Comparison of Result to Traditional Quantum Simulators

In this section, we align the results of our novel implementation with those obtained from Qiskit's entire suite of simulators. Our aim is to verify that our results are consistent with those of well-established, widely-used simulators.

2.1 Chi-Square Goodness-of-Fit Test

To compare our results with those of the extant quantum simulators, we deployed the Chi-Square Goodness-of-Fit Test, a variant of Pearson's chi-square test. This test is employed to determine if the observed distribution of a particular execution result, which is treated as a discrete random variable, deviates from the expected distribution. This expectation is derived from the result of the result of our implementation.

For the reliability of a Chi-Square goodness-of-fit test, a widely accepted rule of thumb stipulates that the expected frequencies should be no less than 5 for all potential outputs. Consequently, we adhered to this guideline in conducting our experiment.

The p-value of the Chi-Square goodness-of-fit test is the probability of obtaining statistical results at least as extreme as the results actually observed, under the assumption that the null hypothesis is correct. A smaller p-value indicates that the observed data is less likely under the null hypothesis, and at a certain threshold (often 0.05), we might reject the null hypothesis in favor of the alternative hypothesis.

In our comparative analysis between the results of our implementation and those of Qiskit's simulators, we formed the null hypothesis stating, "the outcomes of the Qiskit simulator conform to the probability distribution derived from our implementation". The Chi-Square goodness-of-fit test was consequently conducted under this presupposition.

2.2 Compared Simulators

We conducted a comparison of the results from twelve quantum simulators provided by Qiskit to determine whether they align with the probability distribution derived from our implementation.

Quantum Simulator	Explanation	Dynamic Circuit Support
aer_simulator	Explanation for Simulator 1	?
aer_simulator_statevector	Explanation for Simulator 2	?
aer_simulator_density_matrix	Explanation for Simulator 3	?
aer_simulator_stabilizer	Explanation for Simulator 4	?
aer_simulator_matrix_product_state	Explanation for Simulator 5	?
aer_simulator_extended_stabilizer	Explanation for Simulator 6	?
aer_simulator_unitary	Explanation for Simulator 7	?
aer_simulator_superop	Explanation for Simulator 8	?
qasm_simulator	Explanation for Simulator 9	?
statevector_simulator	Explanation for Simulator 10	?
unitary_simulator	Explanation for Simulator 11	?
pulse_simulator	Explanation for Simulator 12	?

Table 1: Qiskit Quantum Simulators

2.3 Benchmarks

To compare the behaviors of both static and dynamic quantum circuits between our implementation and Qiskit Aer simulators, we executed a total of 33 benchmark programs from QASMBench[ ] and all examples found in the OpenQASM2.0 specification.

2.3.1 QASMBench

We executed all the small-scale benchmarks specified in the QASMBench paper along with some of the medium-scale benchmarks that involve 8 qubits or less, using our implementation (refer to Table 2). This enabled us to derive the expected probability distribution for each benchmark program.

2.3.2 OpenQASM2.0 Specification

All the benchmark programs listed in Table 2 are static circuits. To facilitate a comparison of dynamic circuit behaviors, we incorporated dynamic circuit benchmarks as introduced in the OpenQASM2.0 specification (See Table 3).

2.4 Experiment Result

Presented below are the p-values derived from the chi-square goodness-of-fit test for all twelve simulators, as compared to our implementation using all 33 benchmarks.
Image (Currently under experiment)

Even after conducting 10 million shots, the p-values remain sufficiently large for us to confidently accept the null hypothesis asserting conformance between the simulator and our implementation.
There were occaional instances where we observed p-values smaller than 0.05. However, careful and rigorous evaluation is necessary to confirm whether these instances truly indicate a physical semantics error. (Confirmed in small-scale experiments)

3 Comparisons to Previous Works

This section emphasizes the potential for utilizing novel approaches for testing quantum simulators that have not yet been explored.

Index	Benchmark	Scale	Qubits	Gates	CX
1	adder	small	4	23	10
2	basis_change	small	3	53	10
3	basis_trotter	small	4	1626	582
4	bell_state	small	2	3	1
5	cat_state	small	4	4	3
6	deutsch	small	2	5	1
7	dnn	small	2	268	84
8	fredkin_n3	small	3	19	9
9	qec_dist3	small	5	114	49
10	grover	small	2	16	2
11	hs4	small	4	28	4
12	inverseqft	small	4	8	0
13	iSWAP	small	2	9	2
14	linearsolver	small	3	19	4
15	lpn	small	5	11	2
16	pea	small	5	98	42
17	qaoa	small	3	15	6
18	qec_sm	small	5	5	4
19	qec_en	small	5	25	10
20	qft	small	4	36	12
21	qrng	small	4	4	0
22	quantumwalks	small	2	11	3
23	shor	small	5	64	30
24	toffoli	small	3	18	6
25	teleportation	small	3	8	2
26	jellium	small	4	54	16
27	vqe	small	4	89	9
27	vqe_uccsd	small	4	220	88
28	wstate	small	3	30	9
29	dnn	medium	8	1200	384
30	bb84	medium	8	27	0
31	qaoa	medium	6	270	54
32	simons	medium	6	44	14
33	hhl	medium	7	689	196

Table 2: QASMBench

3.1 Problems

Earlier methods for testing quantum simulators, such as QDiff and MorphQ, have employed two-sample tests, specifically the Kolmogorov-Smirnov test (K-S test). However, there are two key issues with this approach.

1. Two-sample testing is performed without the ideal probability distribution that the simulation results should align with, which inherently decreases the test's accuracy.

2. The K-S test is fundamentally a statistical test for a continuous probability distribution, which is not a suitable fit for our case where we are dealing with a discrete probability distribution. QDiff employs a variant of the K-S test for discrete instances, but they don't compute the p-value for the test; instead, they use it to determine "*how many measurements are needed for a reliable evaluation to ensure the relative error between two distributions is within a given threshold t with confidence p?*" On the other hand, MorphQ computes the p-value using the standard K-S test for a continuous probability distribution without any modifications, but the relevance of the p-value obtained this way remains ambiguous.

Index	Benchmark	S/D	Qubits	Gates	CX
1	inverse QFT followed by measurement	dynamic	?	?	?
2	quantum error correction	dynamic	?	?	?
3	quantum fourier transform	static	?	?	?
4	quantum process tomography	static	?	?	?
5	quantum teleportation	dynamic	?	?	?
6	randomized benchmarking	static	?	?	?
7	ripple carry adder	static	?	?	?

Table 3: OpenQASM2.0 Specification Examples

3.2 One-sample testing

The results produced by our implementation, represented as a discrete probability distribution, enable the use of one-sample testing, effectively addressing both aforementioned issues.

1. As we have knowledge of the correct expected probability distribution, we can conduct one-sample testing that is more capable of identifying semantic errors.

2. The Chi-square goodness-of-fit test is designed for discrete probability distributions and is more suited to our case. It provides a p-value, essentially answering the question, "Does it make sense to say that this distribution originated from this particular probability distribution?".

3.3 Effectiveness Comparison Between One-sample and Two-sample Testing

To verify that one-sample testing with the Chi-square goodness-of-fit test is more proficient at identifying semantic errors compared to two-sample testing with the F-S test, we conducted an experiment examining each method's ability to identify semantic errors when the semantics of the Hadamard gate are incorrect:

```
gate h a { U(pi/2 - 0.1, 0,pi) a; } // incorrect: - 0.1
```

Graph: As the number of shots increases, the count of p-values less than 0.05 also rises. The count of p-values of the one-sample test increases much quicker. Confirmed in small-scale experiments; this experiment is ongoing.

The graph above demonstrates how the error detection rate, based on p-value, increases with the number of shots when testing a benchmark suite with incorrect semantics for the Hadamard gate. For the two-sample test, we used MorphQ's methodology to compute the p-value. We can observe that the one-sample test identifies a significantly higher number of semantic errors.
Thus, testing a quantum circuit smulator, using our implementation will be more effective than the existing method.