

Index

- ⊕ symbol, controlled qubit in quantum circuit notation, 54
- ⊗ tensor product operator, 5
 - Kronecker product operator, 19
 - tensoring together matrices, 13
- ψ as qubit state space, 15
 - operator applied to ψ at qubit index, 33
- ★ operator for tensor products, 19
 - Kronecker products, 13
- @ operator for matrix multiplication, 13
- 9-qubit Shor error correction code, 362, 374
- Absolute value of complex numbers, 1
- Addition
 - constants, 270, 275
 - increment operator, 295, 303
 - quantum arithmetic, 265–271, 276
 - quantum gates, 265–272
 - testing quantum arithmetic, 270, 275
- `adjoint()` function for gates, 30, 56
 - qc data structure, 82
- ALAP scheduling, 379, 393
- Amdahl's law, 359, 372
- Amplitude amplification
 - amplitude estimation, 224–228
 - Boolean Satisfiability, 228–232
 - graph coloring, 232–236
 - Grover's algorithm, *see* Grover's algorithm
 - quantum amplitude amplification, 218–221
 - quantum counting, 222–224
 - quantum mean estimation, 237–239
 - quantum median estimation, 241–243
 - quantum minimum finding, 239–241
- Amplitude damping, 355, 367
- Amplitude encoding, 178
- Amplitude estimation, 222, 224–228
 - quantum counting, 222–224
- Ancilla qubits (ancillae), 57
 - code to create and initialize registers, 79
 - compiler optimization and, 375, 390
 - entanglement, 58, 67
 - error correction trick, 359, 371
 - quantum computation, 67
 - Quipper programming language, 373, 387
 - Silq programming language, 374, 388
 - uncomputation, 66
- AND logic gates, 126
- Ansatz, 305, 314
- Arithmetic via quantum gates
 - decrement circuit, 295, 303
 - full adder, 123–126
 - code, 124
 - constants, 270, 275
 - quantum arithmetic, 265–271, 276
 - increment circuit, 295, 303
 - multiplication, 269, 271, 274, 276
 - powers, 289, 297
 - testing quantum arithmetic, 270, 275
- Array `ndarray` as Tensor base, 12
- Arute, Frank, 128
- at (@) operator for matrix multiplication, 13
- Basis encoding, 177
- Basis states of qubits, 15
 - basis encoding, 177
 - constructing a qubit, 16
 - density matrix diagonal elements, 28, 106
 - measurement, 70
 - orthonormal set of basis vectors, 15
 - superposition as orthonormal basis, 40
 - projection operators extracting
 - amplitude, 46
 - state as superposition, 19
 - superposition via Hadamard gates, 40
 - Hadamard basis, 40
- Bell measurement, 139, 145
- Bell states, 61
 - code, 62
 - measurement example, 74
 - Quipper programming language, 373, 387
 - tracing out qubits, 110
- Bell, John S., 58, 61

- Benchmarking
 - benchmark gaming, 129
 - cross entropy benchmarking, 129
 - gate faster application in C++, 99
 - quantum random circuits, 129
 - quantum versus classical computers, 129–131
 - sparse representation, 103
- Bernstein–Vazirani algorithm, 163–166
 - about, 162
 - oracle form of algorithm, 164, 173
 - compiler optimization, 380, 394
 - phase-kick rotation gates, 245, 246
- Beyond Classical
 - classical arithmetic via quantum gates, 123–126
 - computational complexity theory, 76, 128, 136
 - Google Sycamore processor, 129
 - benchmarking, 129
 - benchmarking quantum versus classical computers, 129–131
 - logic circuit general construction, 126
 - quantum random circuits, 129
 - simulation design, 131
 - simulation evaluation, 135
 - simulation implementation, 133
 - simulation metric, 133
 - Quantum Supremacy experiment, 127–136
- Binary fractions, 24
- Binary interpretation, 23–25
- Birthday paradox, 275, 280
- Bit conversion, 23
- Bit index notation for states, 94
- Bit order
 - binary interpretation, 23–25
 - qubit order, 21–22
 - two tensored states, 22
- Bit-flip errors
 - bit-flip channel, 354, 366
 - bit-flip phase-flip channel, 354, 366
 - combined phase/bit-flip error, 352, 364
 - dissipation-induced error, 351, 363
 - error correction, 359–361, 371–373
 - Shor’s 9-qubit code, 362, 374
- Bits to binary fractions, 24
- Bits-decimal conversion functions, 23
- `bits2frac()` for binary fractions, 24
- `bits2val()` for binary to decimal, 23
- `bitstring()` function, 27
- Black-box algorithms
 - about, 162
 - quantum parallelism, 162
 - query complexity, 162
 - Bernstein–Vazirani algorithm, 163–166
 - oracle form of algorithm, 164, 173
 - Deutsch algorithm, 166–173
 - general oracle operator, 171
 - Deutsch–Jozsa algorithm, 174–176
- Bloch sphere
 - Quirk online simulator, 264, 269
- Bloch spheres
 - about, 17, 38, 302, 312
 - Bloch vector, 18
 - rotation operators, 38
 - expectation values, 17, 71, 303, 312
 - minus sign as global phase, 17
 - qubits described by, 17–19
 - computing coordinates for given state, 37
 - Solovay–Kitaev algorithm, 189
 - two degrees of freedom for superdense coding, 143
 - universal gates, 189
- Bloch, Felix, 17
- Boolean Conjunction, 228
- Boolean Disjunction, 228
- Boolean formulas with quantum gates, 127
- Boolean Satisfiability, 228–232
- Born rule, 69
 - about projective measurement, 69
- BPP (complexity), 128, 136
- BQP (complexity), 128, 136
- Bra(c)kets, 3
- Branching, *see* Controlled gates
- Bras
 - Dirac notation, 2
 - inner products, 3
 - bra-ket notation, 3
 - tensor products, 5
- C++
 - Accelerated gate application
 - Execution speed, 100
 - accelerated gate application, 95–101
 - execution speed, 95
 - extending Python with, 96
 - sparse representation, 101
 - benchmarking, 103
 - “Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?” (Einstein, Podolsky, and Rosen), 58
 - Cartesian coordinates for Bloch sphere, 38
 - CCX gates, *see* controlled–controlled Not gates
 - Change of basis, 342, 354
 - Channels in information theory, 353, 365
 - bit-flip channel, 354, 366
 - bit-flip phase-flip channel, 354, 366
 - depolarization channel, 354, 366
 - phase-flip channel, 354, 366
 - ChatGPT, 331, 342
 - CHSH game, 146–149

- Circuits
 - about function calls and returns, 370, 384
 - Scaffold programming language, 370, 384
 - Silq programming language, 374, 388
 - compiler optimization and, 376, 390
 - decrement circuit, 295, 303
 - entangler circuits, 59–61
 - increment circuit, 295, 303
 - intermediate representation
 - capabilities, 88
 - subcircuit control, 89
 - libraries of compiler optimization
 - patterns, 380, 394
 - Logic circuits
 - fan-out circuits, 126
 - logic circuits, 126
 - fan-out in QCL, 368, 382
 - phase inversion circuit, 209
 - quantum circuit data structure, 80
 - adjoint gates, 82
 - constructor, 80
 - double-controlled gates, 83
 - gates, 82–85
 - gates applied, 82
 - measurements, 86
 - multi-controlled gates, 84
 - parameterized gates, 83
 - quantum registers, 81
 - qubits added, 81
 - Swap and controlled Swap gates, 85
 - quantum random circuits, 129
 - simulation design, 131
 - simulation evaluation, 135
 - simulation implementation, 133
 - simulation metric, 133
 - qubits
 - ordering of qubits, 22
 - qc data structure, 81
 - quantum circuit notation, 53–55
 - Shor’s 9-qubit error correction, 362, 374
 - subcircuits instantiated, not called, 370, 384
- Cirq commercial system (Google), 375, 389
 - simulators, 383, 397
- Classical arithmetic, *see* Arithmetic via quantum gates
- Classical computers versus quantum, 129–131
 - classical computers controlling quantum, 365, 366, 378, 379
 - Summit simulating quantum random circuits, 135
- Clifford gates, 78
- Closed quantum systems, 352, 364
- CNF, 228
- CNOT*, *see* Controlled Not gates
- CNOT0*, *see* Controlled-by-0 Not gate
- Coin toss operator, 294, 302
- Coloring graphs, 232–236
- Column vectors
 - inner products, 3
 - kets
 - Dirac notation, 2
 - Hermitian conjugate of, 2
 - qubits and states as, 2
- Combined phase/bit-flip error, 352, 364
- Commercial systems, 375, 389
- Compiler optimization, 375, 390
 - about, 375, 390
 - classical for classical constructs, 376, 391
 - gate approximation, 382, 396
 - gate elimination, 377, 391
 - gate fusion, 378, 392
 - gate parallelization, 379, 393
 - gate scheduling, 378, 392
 - high-performance pattern libraries, 380, 394
 - inlining, 376, 391
 - logical to physical mapping, 376, 380, 390, 395
 - resource for information, 382, 396
 - loop unrolling, 376, 391
 - peephole optimization, 379, 393
 - libraries of compiler optimization
 - patterns, 380, 394
 - relaxed peephole optimization, 379, 394
 - physical gate decomposition, 381, 396
 - unentangled qubits, 378, 392
- Compilers
 - about hierarchy of abstractions, 367, 380
 - design challenges, 364, 378, 380, 395
 - optimization, *see* Compiler optimization
 - optimization, *see* Compiler optimization
 - programming languages, *see* Quantum programming languages
 - programming languages, *see* Quantum programming languages
- Completeness relation, 71
- Complex numbers
 - numpy data types, 12
 - 2D plane, 1
 - about, 1
 - conjugates, 1
 - exponentiation, 2
 - modulus, 1
 - norm, 1
 - Python, 2
 - qubits as column vectors of, 2

- states as column vectors of, 2
- Tensor comparisons to values, 14
- tensor_type() abstraction, 12
- Complex plane, 1
- Complexity classes
 - BPP, 128, 136
 - BQP, 128, 136
 - NP, 128
 - NP-complete, 128
 - NP-hard, 128
 - P, 128
- Complexity of simulation, 76, 129
- Composite kets inner products, 6
- Computation reversed, 66
- Computational basis, 15
- Computational complexity theory, 76, 128, 136
- Conditional execution, *see* Controlled gates
- Conjugates
 - adjoint synonymous with conjugate, 7
 - complex numbers, 1
 - denotation not explicit, 3
 - Hermitian conjugate matrix, 7
 - operator adjoint() function, 30
- Conjugation
 - conjugate complex numbers, 2
 - involutivity, 2
- Conjunction, 228
- Constants in quantum addition, 270, 275
- Continued fractions, 290, 297
- Controlled U gates under compiler
 - optimization, 379, 394
- Controlled Z gates, 52
- Controlled gates, 48–50
 - about QCL programming language, 369, 382
 - controlled $U_1(\lambda)$ gate for quantum arithmetic, 266, 270
 - Controlled Z gates, 52
 - Controlled Not gates, 48–50
 - constructor function, 50
 - Controlled phase gates, 52
 - controlled rotation gates additive, 244
 - controlled–controlled gates, 50
 - qc data structure, 83
 - Toffoli gates, 55, 56
 - Controlled-by-0 Not gate, 50
 - function of, 48–50
 - multi-controlled gates, 57
 - ancilla qubits, 57, 66
 - controlled–controlled Not gates, 55
 - qc data structure, 84
 - Sleator–Weinfurter construction, 56
 - nonadjacent controller and controlled qubits, 49
 - notation for gates involved, 48
 - qc data structure
 - double-controlled gates, 83
 - fast application of gates, 94
 - multi-controlled gates, 84
 - Swap and controlled Swap gates, 85
 - quantum circuit notation
 - controlled X gates, 54
 - controlled Z gates, 54
 - Controlled-by-0 Not gate built, 54
 - more than one qubit controlling, 54
 - scalability, 50
 - Swap and controlled Swap gates
 - qc data structure, 85
- Swap gates, 51
 - compiler optimization, 379, 394
 - controlled Swap gates, 51
 - quantum circuit notation, 54
- Controlled Not gates ($CNOT$; CX)
 - compiler optimization, 380, 394
 - constructor function, 50
 - entangler circuits, 60
 - function, 48–50
 - GHZ states, 62
 - logic circuits from, 126
 - quantum registers for result storage, 68
 - Swap gate action, 51
- Controlled phase gates, 52
- Controlled Swap gates, 51
- Controlled–Controlled gates
 - Toffoli gates, 55
- Controlled–Controlled gates
 - Sleator–Weinfurter construction, 56
 - Toffoli gates
 - logic circuits from, 126
 - Sleator–Weinfurter construction, 56
- Controlled–Controlled Not gates (CCX gates), 55
- Controlled-by-0 Not gate ($CNOT0$), 50
 - quantum circuit notation, 54
- Copenhagen interpretation of quantum mechanics, 59
- Counters
 - decrement operator, 295, 303
 - increment operator, 295, 303
- Covariance matrix, 332, 343
- C++
 - Q language C++ class library, 372, 386
- cQASM, 367, 381
- Cross entropy benchmarking (XEB), 129
- Curse of dimensionality, 331, 342
- Cut on graph, 315, 325
- CX , *see* Controlled Not gates
- Data encoding
 - amplitude encoding, 178
 - basis encoding, 177
 - Hamiltonian encoding, 180
 - rotations for encoding, 179
- Data registers, *see* Quantum registers

- Data structure, *see* Quantum circuit (qc)
 - data structure
- Data types
 - about, 12
 - abstracting, 12
 - complex data type selection, 12
 - numpy data types, 12
 - State, 16
- Debugging
 - direction of rotations, 42
 - qc data structure abstraction, 80
 - Tensors compared to values, 14
- Decoherence times of technologies, 350, 362
- Decoherence-induced phase shift error, 351, 363
- Decrement circuit, 295, 303
- Density matrices
 - about, 28
 - apply gate, 107
 - Bloch sphere coordinate computation
 - Cartesian coordinates, 38
 - outer product of state with itself, 28, 106
 - partial trace derivation
 - code, 108
 - tracing out other qubits, 109
 - probabilities of measuring a basis state, 28
 - quantum computing theory as, 106
 - as tool, 106
 - trace of, 28, 107
- Depolarization channel, 354, 366
 - depolarization definition, 354, 366
- Destructive interference, 343, 355
- Deutsch algorithm, 166–173
 - about, 162
 - general oracle operator, 171
- Deutsch–Jozsa algorithm, 174–176
- Diagonal matrices
 - eigenvalues, 6
 - tensor products, 5
- Diffusion operator, 210
- Dirac notation
 - bras, 2
 - kets, 2
 - qubits
 - 0-state and 1-state, 15
 - two tensored states, 22
- Discrete phase gates, 42
- Disjunction, 228
- Dissipation-induced error, 351, 363
- Dot products, *see* Inner products
- Double-controlled gates
 - qc data structure, 83
- Dual vectors for a ket, 2
- Dumper function, 26
 - transpilation, 90
- EGCD, *see* Extended Euclidean algorithm
- Eigenstates, 6
- Eigenvalues, 6
 - Hamiltonians, 300–301, 308–310, 317–319
 - Hermitian and Pauli matrices, 37
 - Hermitian matrices, 7
 - quantum phase estimation, 247, 248
 - trace of a matrix, 9
 - unitary matrices, 8
 - variational quantum eigensolver, 308–309, 317–319
- Eigenvectors, 6
 - Hamiltonians in Schrödinger equation, 300–301, 309–310
 - unitary matrices, 8
- Einstein, Albert
 - hidden state, 58
 - spooky action at a distance, 58, 61
- Electron decoherence time
 - electron spin, 350, 362
 - gallium arsenide, 350, 362
 - gold, 350, 362
- Embeddings, 331, 342
- Encoding data, *see* Data encoding
- Endianness of qubits
 - quantum circuit notation, 53
- Endianness of qubits, 21–22
- Entanglement, 58–65
 - about, 58
 - algorithms exploiting
 - CHSH game, 146–149
 - entanglement swapping, 145
 - random number generator, 137
 - superdense coding, 142–145
 - teleportation, 138–142
 - analysis by Scaffold, 371, 385
 - ancilla qubits, 58, 67
- Bell states, 61
 - code, 62
 - tracing out qubits, 110
- code
 - Bell states, 62
 - entangler circuit, 60
 - GHZ states, 62
- compiler optimization and, 378, 392
- Copenhagen interpretation, 59
- entangler circuits, 59–61
 - code, 60
- GHZ states, 62
 - code, 62
 - error correction trick, 359, 371
- maximal entanglement, 111
- mixed state depolarization, 354, 366
- No-Cloning Theorem, 64
 - error correction challenge, 358, 370

- product states, 59
- tracing out qubits, 110
- W state, 63
- Entangler circuits, 59–61
- Environmental challenges of quantum computing, 350–357, 362–369
 - closed versus open quantum systems, 352, 364
- EPR paper, 58
- Equal superposition of adjacent qubits, 41
- Erasure of information resulting in heat dissipation, 66
- Error correction
 - about, 350, 357, 362, 369
 - bit-flip errors, 359–361, 371–373
 - Shor's 9-qubit code, 362, 374
 - compiler optimization and, 376, 390
 - error correction code memory, 357, 369
 - error syndrome, 359, 371
 - phase-flip errors, 361, 373
 - Shor's 9-qubit code, 362, 374
 - quantum computing challenges, 357, 370
 - quantum noise, 350–357, 362–369
 - repetition code, 357, 369
 - majority voting, 357, 369
 - No-Cloning Theorem, 358, 370
 - quantum repetition code, 358, 370
 - resources for information, 363, 376
 - Shor's 9-qubit error correction code, 362, 374
- Error correction code memory (ECC), 357, 369
- Error injection to model quantum noise, 355, 367
 - checking bit-flip error correction, 359, 372
 - gates as quantum noise source, 356, 368
- Error syndrome, 359, 371
- Euclidean distance, 327–331, 338–342
 - quantum algorithms that use, 331, 342
- Euler theorem, 276, 281
- Euler's formula
 - complex exponentiation, 2
 - Phase gate derivation, 41
- Expectation values, 17, 71
 - Bloch sphere, 17
 - variational quantum eigensolver, 303, 312
- Exponentiation
 - complex numbers, 2
 - operators, 38
- Extended Euclidean algorithm, 285, 292
- Factorization, 274, 279
- Fan-out circuits, 126
 - QCL programming language, 368, 382
- fast gate application, 92–95
- faster gate application, 95–98
- Feynman, R., ix, xii
- Flexible phase gates
 - constructed via U_3 gates, 43
 - constructing other gates, 43
 - discrete phase gates, 42
 - $U_1(\lambda)$ gates, 43
- Fourier transform, *see* Quantum Fourier transform (QFT), *see* Quantum Fourier transform (QFT)
- Fractions, binary, 24
- Fredkin gates, 51
- Full adder, 123–126
 - code, 124
 - quantum arithmetic, 265–271, 276
- Fundamentals of quantum computing, *see* Quantum computing fundamentals
- Fused gates, 378, 392
- Gallium arsenide (GaAs) electron decoherence time, 350, 362
- Gate equivalences
 - compiler optimization, 378, 380, 392, 395
 - Controlled phase gates, 52
 - multi-controlled gates, 57
- Gates
 - qc data structure
 - parameterized gates, 83
 - about operators as gates, 29
 - adjoint gate qc data structure, 82
 - application, 30–31
 - `apply()` function, 34, 82, 98
 - fast application, 92–95
 - fast application generalized, 94–95
 - faster application with C++, 95–101
 - fastest benchmarked, 103
 - fastest with sparse representation, 101–102
 - multiple operators in sequence, 33
 - multiple qubits, 31–33
 - noise reduction via compiler optimization, 376, 390
 - norm preserving, 29
 - notation for qubit index applied to, 33
 - padding operators, 33, 49
 - projection operators extracting subspace, 70
 - quantum computation, 67
 - to density matrix, 107
 - to state ψ at qubit index, 33
- compiler optimization
 - gate approximation, 382, 396
 - gate fusion, 378, 392
 - gate parallelization, 379, 393
 - gate scheduling, 378, 392

- Gates (*conti.*)
 logical to physical mapping, 376, 380, 390, 395
 noise reduction, 376, 390
 physical gate decomposition, 381, 396
 scheduling gates as late as possible, 379, 393
 unentangled qubits, 378, 392
 constructed via U_3 gates, 43
 controlled gates, *see* Controlled gates
 flexible phase gates
 discrete phase gates, 42
 phase shift or kick gate, 43
 Hadamard gates, 40–41
 identity gates, 35
 applied to multiple qubits, 32
 multi-qubit gates
 controlled gates, 48–55
 Hadamard gates, 40–41
 single-qubit constructors for, 35
 outer product representation of operator, 46
 parameterized gate quantum circuit data structure, 83
 phase gates, 41
 discrete phase gates, 42
 phase inversion operator, 209
 phase shift or kick gates, 43
 square root as T gate, 45
 various gates via, 43
 projection operators, 46
 qC data structure, 82–85
 double-controlled gates, 83
 gates applied, 82
 multi-controlled gates, 84
 quantum circuit notation, 53
 quantum noise source, 356, 368
 precision of design required, 365, 378
 R_k gates, 42
 Scaffold programming language, 370, 383
 Classical-To-Quantum-Circuit tool, 370, 384
 single-qubit gates, 34–41
 Solovay-Kitaev theorem, 188
 T gates
 square root of S gates, 45
 universal gates, 189
 via phase gates, 43
 $U_1(\lambda)$ gates, 43
 universal gates, *see* Universal gates
 V gates as square roots of X gates, 44, 56
 X gates, 29, 36
 square root of as V gates, 56
 Y gates, 36
 square root of, 45
 yroot gates, 45
 GCD, *see* Greatest common divisor, *see* Greatest common divisor
 GHZ states, 62
 error correction trick, 359, 371
 Global phase, 17
 Bloch sphere, 17
 phase invariance, 17
 Global variables as bad style, 28
 “Going beyond Bell’s Theorem” (Greenberger, Horne, and Zeilinger), 62
 Gold (Au) electron decoherence time, 350, 362
 Google
 Cirq commercial system, 375, 389
 simulators, 383, 397
 coding style, 14
 underscore in function names, 14
 quantum random circuits, 129, 135
 simulation design, 131
 simulation evaluation, 135
 simulation implementation, 133
 simulation metric, 133
 Sycamore processor supremacy, 129–131
 GPUs (graphics processing units), 364, 378
 Gradient descent, 307, 316
 Graph coloring, 232–236
 Graph cut, 315, 325
 graphics processing units (GPUs), 364, 378
 Greatest common divisor (GCD), 274, 279
 Greenberger, Daniel M., 62
 Ground state energy
 about variational quantum eigensolver, 299, 308
 variational principle, 301, 310
 Grover’s algorithm
 about, 200
 accounting for multiple solutions, 218–221
 circuit implementation, 216–218
 examples
 simple numerical, 203
 two-qubit, 204
 Grover operator, 201, 207
 implementing, 215
 quantum counting, 222
 inversion about the mean, 203
 circuit, 213
 operator, 209–213
 iteration count, 206–208
 multiple solutions, 218–221
 overview, 201
 phase inversion, 201

- implementation, 208
 - multiple solutions, 218–221
 - operator, 209, 222
- quantum amplitude amplification, 218–221
- quantum counting, 222–224
- Hadamard basis, 15, 18, 40
 - measuring in, 144
- Hadamard gates, 40–41
 - constructed with U_3 gates, 44
 - entangler circuits, 59–61
 - Hadamard basis, 40
 - measuring in, 144
 - Hadamard coin, 294, 302
 - its own inverse, 41
 - quantum circuit notation, 53
 - random number generator, 137
 - universal gates, 189
- Hadamard similarity test, 155–160
- Hamiltonian
 - definition, 300, 309
 - eigenvalues
 - about VQE algorithm, 299, 302, 308, 311
 - Schrödinger equation derivation, 300–301, 309–310
 - variational principle, 301, 310
 - variational principle measurements, 308–309, 317–319
 - Ising spin glass model, 314, 324
 - Hamiltonian constructed, 318–320, 328–331
 - operator, 300, 310
 - Hermitian, 301, 310
- Hamiltonian encoding, 180, 342, 354
- Hash table in libq, 388, 393, 403, 409
- Haskell programming language, 372, 386
 - Quipper as embedded DSL, 372, 386
 - oracle construction, 373, 387
 - Silq as embedded DSL, 374, 388
 - oracle construction, 374, 388
- Heisenberg uncertainty principle, 300, 310
- Hello World for quantum computing, 137
- Helper functions
 - bit conversion, 23
 - Bloch sphere coordinate computation, 38
 - Hermitian and unitary matrix
 - properties, 14
 - n -bit projector construction, 47
- Hermitian conjugate vector, 2
- Hermitian matrices
 - about, 7
 - checking if Tensor is Hermitian, 14
 - eigenvalues as real, 7
 - Hermitian adjoint matrices, 7
 - expressions, 8
 - Hermitian conjugate matrices, 7
 - projection operators as, 46
 - real vector space basis, 37
- Hermitian projector, 46
- Hidden state, 58, 61
- Hierarchical QASM, 371, 384
- High-Performance Computing (HPC)
 - techniques, 77
- Horne, Michael A., 62
- I matrix, 7
- IBM
 - Qiskit commercial system, 375, 389
 - ALAP scheduling of gates, 379, 393
 - simulators, 383, 398
 - Sycamore supremacy challenged, 129
 - Summit supercomputer, 135
- Idempotent projection operators, 46
- Identity gates, 35
 - applied to multiple qubits, 32
 - controller and controlled qubits not adjacent, 49
 - Hermitian matrix real vector space, 37
 - via phase gates, 43
- Identity matrix, 7
- Increment circuit, 295, 303
- Increment modulo 9 circuit, 295, 304
- Indirect measures of similarity between
 - states
 - swap test, 150–154
 - swap test code, 153
 - swap test for multi-qubit states, 154
- Information
 - erasure resulting in heat dissipation, 66
 - quantum circuit double lines, 54
 - quantum teleportation, 138–142
 - superdense coding, 142–145
- Inner products, 3
 - tensors, 6
- Instruction Set Architecture (ISA) of
 - quantum computers, 29
- Intel Quantum Simulator, 382, 397
- Intermediate representation (IR), 86–91
 - about circuit capabilities, 88
 - Scaffold programming language, 370, 384
 - classic and quantum mix, 376, 391
 - scalability, 365, 378
- transpilation
 - dumper function, 90
 - inverting a register, 90
 - IR base class, 87
 - IR nodes, 86
 - subcircuit control, 89
 - uncomputation, 88
- Inversion about the mean, 203
 - circuit, 213
 - operator, 209–213

- Inversion test for similarity, 160
- Involutivity, 2
 - Hadamard gates, 41
 - Pauli matrices, 37
 - rotations, 38
- Ion trap decoherence time, 350, 362
- IR, *see* Intermediate representation
- ISA (Instruction Set Architecture) of quantum computers, 29
- Ising
 - Hamiltonian, 314, 324
 - NP algorithms, 314, 324
 - Spin Glass, 314, 324
- Junk qubits, 66
 - quantum computation, 67
- K-nearest neighbor algorithm, 331, 342
- KD-Tree, 192
- Kets
 - about, 68
 - composite kets inner products, 6
 - Dirac notation, 2
 - dual vectors for, 2
 - Hermitian conjugate of, 2
 - inner products, 3
 - bra-ket notation, 3
 - composite kets, 6
 - outer products, 4
 - trace of, 9
 - tensor products, 5
- KNN, 331, 342
- Knuth, D. E., ix, xii
- Kraus operators, 71, 353, 365
- kron member function of Tensor, 13
- Kronecker power function (kpow), 13
- Kronecker product, 5, 13
 - $\otimes$ operator symbol, 5, 19
 - $\star$ operator for, 13, 19
 - tensor product synonym, *see also* Tensor products
- Landauer, D., 66
- Landauer's principle, 66
- Least significant bit, *see* Bit order
- Libq, 101
 - implementation
 - about, 384, 399
 - controlled gates, 387, 402
 - gate application, 390–392, 406–408
 - hash table, 388, 393, 403, 409
 - register file, 384, 399
 - superposition-preserving gates, 385, 400
 - superpositioning gates, 388, 403
 - libquantum basis, 101
 - optimization
 - gate application, 393, 409
 - hash table reconstruction, 393–395, 410–411
- libquantum library for sparse representation, 101
- simulation, 382, 397
- Libraries of compiler optimization patterns, 380, 394
- Linear independent vectors, 4
- LLM, 331, 342
- Local phase, 17
- Logic circuits, 126
 - fan-out circuits, 126
 - QCL programming language, 368, 382
- “Logical Reversibility of Computation” (Bennett), 66
- Majority voting for repetition code, 357, 369
- Matrices
 - $\star$ operator for Kronecker product, 13
 - @ operator for matrix multiplication, 13
 - 2-dimensional index via projection operators, 46
 - density matrices, 28, 106
 - diagonalization function, 196
 - eigenvalues, 6
 - exponent with matrix, 38
 - Hermitian, *see* Hermitian matrices
 - Pauli matrices, 35
 - Hermitian matrix real vector space, 37
 - involutivity, 37
 - permutation matrices, 14
 - scalability, 76
 - tensor products, 5
 - tensoring together with $\otimes$, 13
 - trace of, 107
 - trace of a matrix, 9
 - transposition, 2
 - unitary, 7
- Maximal entanglement, 111
- Maximally mixed state, 111
- Maximum cut algorithm, 314–322, 324–333
 - about, 314, 324
 - cut definition, 315, 325
 - experiments, 320, 331
 - Ising formulations of NP algorithms, 314, 324
 - maximum cut definition, 315, 325
 - quantum approximate optimization algorithm, 312, 322
 - variational quantum eigensolver VQE by peek-a-boo, 320, 331

- weighted maximum cut, 315, 325
 - computing maximum cut, 316, 326
 - graphs constructed, 315, 325
 - Hamiltonian constructed, 318–320, 328–331
- Mean estimation, 237–239
- Mean inversion, *see* Inversion about the mean
- Measurement gate quantum circuit notation, 54
- Measurements
 - by peek-a-boo, 86
 - Grover's algorithm, 215
 - By peek-a-boo, Grover's algorithm, 215
 - entanglement, 58
 - No-Cloning Theorem, 64
 - error detection challenges, 358, 370
 - expectation values, 17, 71, 303, 312
 - Hadamard basis for measuring, 144
 - implementation, 72
 - Pauli bases, 302–305, 312–314
 - projective, 69–72
 - examples, 73
 - implementation, 72
 - q/c data structure, 86
 - quantum circuit notation, 54
 - quantum mechanics postulates, 68
 - state similarity indirect measures
 - swap test, 150–154
 - swap test code, 153
 - swap test for multi-qubit states, 154
 - states collapsing on measurement, 16, 58
 - Born rule, 69
 - measurement definition, 69
 - renormalization, 72
- Median estimation, 241–243
- Mermin, David, 59
- Microsoft Q# commercial system, 375, 389
 - Quantum Developer Kit, 375, 389
 - simulators, 383, 398
- Microwave cavity decoherence time, 350, 362
- Minimum cut problems, 315, 325
- Minimum spanning tree, 331, 342
- Mixed states
 - depolarization, 354, 366
 - tracing out qubits, 111
- Mixed-product property, 6
- MLPerf benchmarks, 129
- Modular arithmetic, 273, 278
 - continued fractions, 290, 297
 - controlled modular multiplication, 288, 295
 - modular addition, 286–288, 293–295
- Modular inverse, 285, 292
- Modulus of complex numbers, 1
- Most significant bit, *see* Bit order
- Multi-controlled gates, 57
 - ancilla qubits, 57, 66
 - controlled–controlled Not gates, 55
 - q/c data structure, 84
 - Sleator–Weinfurter construction, 56
- Multi-qubit gates
 - about controlled gates, 48
 - about single-qubit constructors, 35
 - Hadamard gates, 40–41
- Multiplication, 269, 274
 - quantum arithmetic, 271, 276
 - testing quantum arithmetic, 270, 275
- Möttönen's algorithm, 182–188
- NAND logic gates, 126
- nbits property of Tensor class, 23
- ndarray base for Tensor, 12
- No-Cloning Theorem, 64
 - fan-out circuits and, 126
 - repetition code for error control, 358, 370
 - uncomputation not violating, 68
- No-Deleting Theorem, 65
- Node class for transpilation, 86
- Noise, *see* Quantum noise, *see* Quantum noise
- Noisy Intermediate Scale Quantum Computers (NISQ), 299, 308, 365, 378
- Norm
 - complex numbers, 1
 - unitary matrices as norm preserving, 7, 29
 - vector normalization, 26
- Not gates, *see also* X gates
 - logic circuits from, 126
- Nuclear spin decoherence time, 350, 362
- numpy
 - path to, 96
- numpy
 - ★ operator for Kronecker product, 13
 - @ operator for matrix multiplication, 13
 - adjoint() function for operators, 30
 - allclose() for Tensor comparisons, 14
 - conj function, 2
 - ndarray base for Tensor, 12
 - instantiating, 12
 - about, 11
 - complex number support, 12
 - eigenvalues of matrices, 7
- “On the Einstein Podolsky Rosen paradox” (Bell), 58, 61
- Open quantum systems, 352, 364
- Open-source simulators, 382, 396

- OpenPulse, 367, 380
- OpenQASM, 367, 381
 - transpilation dumper function, 90
- Operator class
 - `adjoint()` function, 30
 - gate applied to state ψ at qubit index, 34
 - Gate function returning Operator object, 34
 - Tensor class parent, 29
- Operator function, 181
- Operator-sum representation, 353, 365
- Operators
 - $\star$ operator for Kronecker product, 13
 - $\otimes$ operator for matrix multiplication, 13
 - about, 12, 29
 - application, 30–31
 - `apply()` function, 34, 82, 98
 - fast application, 92–95
 - fast application generalized, 94–95
 - faster application with C++, 95–101
 - fastest benchmarked, 103
 - fastest with sparse representation, 101–102
 - multiple operators in sequence, 33
 - multiple qubits, 31–33
 - noise reduction via compiler optimization, 376, 390
 - norm preserving, 29
 - notation for qubit index applied to, 33
 - padding operators, 33, 49
 - projection operators extracting subspace, 70
 - quantum computation, 67
 - to state ψ at qubit index, 33
- diffusion operator, 210
- Hamiltonian operator, 300, 310
 - Hermitian, 301, 310
- inversion about the mean, 209–213
 - circuit, 213
- oracle operator, 171
 - phase inversion implementation, 208
- outer product representation, 46
- Pauli representation, 117–120
- phase inversion operator, 209
 - quantum counting, 222
- q_c data structure, 82
 - gates applied, 82
- quantum Fourier transform operator, 262, 266
 - inverse, 263, 267
- Tensor class parent, 29
- unitary, 29
 - invertable, 29
- Optical cavity decoherence time, 350, 362
- Optimization
 - compilers, *see* Compiler optimization, *see* Compiler optimization
 - gate application iteration lesson, 393, 409
 - gate application special cases, 99–100
 - Hamiltonians constructed for, 314, 324
 - hash table reconstruction, 393–395, 410–411
 - Ising formulations of NP algorithms, 314, 324
 - maximum cut algorithm, 314–322, 324–333
 - quantum approximate optimization algorithm, 312, 322
 - subset sum algorithm, 322–326, 333–337
 - variational quantum eigensolver, 299–312, 322
- OR logic gates, 126
- Oracles
 - about, 162
 - quantum parallelism, 162
 - query complexity, 162
 - Bernstein–Vazirani algorithm, 163–166
 - compiler optimization, 380, 394
 - oracle form of algorithm, 164, 173
 - Deutsch algorithm, 166–173
 - general oracle operator, 171
 - Deutsch–Jozsa algorithm, 174–176
 - general oracle operator, 171
 - phase inversion implementation, 208
 - Quipper automatic construction of, 373, 387
 - RevKit for constructing reversible, 383, 397
 - Silq construction of, 374, 388
- Order finding
 - order of function, 276, 281
 - quantum order, 279, 285
 - quantum algorithm, 279–292, 300
 - continued fractions, 290, 297
 - controlled modular multiplication, 288, 295
 - experimentation, 290, 298
 - main program, 283–284, 290–292
 - modular addition, 286–288, 293–295
 - support routines, 284–286, 292–293
 - Shor’s integer factorization algorithm, 275–277, 280–282
- Orthogonal vectors, 4
- Outer products
 - about, 4
 - density matrices as, 28, 106
 - outer product representation of operator, 46
 - projection operators, 46
 - trace of two kets, 9

- Overloading $\star$ operator, 13
- P gates, *see also* Phase gates
- Parallelism, *see* Quantum parallelism
- Parameterized gate quantum circuit data structure, 83
- Partial-trace procedure
 - code, 108
 - experimenting with, 109
 - dimension reducing operation, 108
 - maximal entanglement, 111
 - reduced density operator from, 107
 - tracing out other qubits, 109
 - entangled states, 110
 - environment traced out, 353, 365
 - experimenting with, 109
 - mixed states, 111
 - Quirk qubits on Bloch sphere, 264, 269
- Path to numpy, 96
- Pauli matrices
 - about, 35
 - Hermitian matrix real vector space, 37
 - involutivity, 37, 39
 - measurement in Pauli bases, 302–305, 312–314
 - Pauli X gates, *see also* X gates
 - Pauli Y gates, 36
 - Pauli Z gates, 36
 - Phase-flip gates, 36
 - quantum noise modeling, 355, 367
 - rotation operators via exponentiation, 38
- Pauli representation, 117–120
 - decomposition with projectors, 119
 - Pauli basis, 117
 - two qubits, 119
- PCA (principal component analysis), 331–336, 342–347
- Peephole optimization, 379, 393
 - libraries of compiler optimization patterns, 380, 394
 - relaxed peephole optimization, 379, 394
- Perdomo two-qubit state preparation, 181
- Performance
 - compiler optimization and, 376, 390
 - quantum versus classical computers, 129–131
- Period of function
 - about, 276, 281
 - quantum order, 279, 285
 - quantum algorithm
 - continued fractions, 290, 297
 - controlled modular multiplication, 288, 295
 - experimentation, 290, 298
 - main program, 283–284, 290–292
 - modular addition, 286–288, 293–295
 - support routines, 284–286, 292–293
 - Shor's integer factorization algorithm, 275–277, 280–282
- Permutation matrices
 - about, 14
 - checking if tensor is permutation, 14
 - Controlled Not gate, 48, 49
- Phase damping, 356, 368
- Phase estimation for π approximation, 256–261
- Phase gates, 41
 - controlled phase gates, 52
 - discrete phase gates, 42
 - phase inversion operator, 209
 - phase shift or kick gates, 43
 - square root of S gate, 45
 - $U_1(\lambda)$ gates, 43
 - controlled $U_1(\lambda)$ gate for quantum arithmetic, 266, 270
 - various gates via, 43
- Phase invariance, 17
- Phase inversion, 201
 - implementation, 208
 - multiple solutions, 218–221
 - operator, 209
 - quantum counting, 222
- Phase of qubits, 26
- Phase shift error, decoherence-induced, 351, 363
- Phase-flip errors, 351, 363
 - bit-flip phase-flip channel, 354, 366
 - error correction, 361, 373
 - Shor's 9-qubit code, 362, 374
 - phase-flip channel, 354, 366
- Phase-kick circuit, 244
- Phase/bit-flip errors, 352, 364
- π approximation via QFT, 256–261
- Planck constant, 300, 310
- Podolsky, B., 58
- Positive operator-valued measure, 71
- Postulates of quantum mechanics, 68
- POVM, 71
- Power arithmetic, 289, 297
- Power function via Kronecker products, 13
- Preskill, John, 127
- Principal component analysis (PCA), 331–336, 342–347
- Probabilistic Turing machines, 128
- Probability amplitudes, 15
 - binary addressing, 23
 - ket definition, 68
 - maximally mixed state, 111
 - measurement, 69, 70
 - qubits as states, 16, 19
 - equal superposition with same amplitude, 41
- Probability amplitudes (*conti.*)

- projection operators extracting amplitude, 46
 - state class code, 25
 - state collapsing on measurement, 16, 69
 - state vectors and unitary operators, 29
 - Swap gates, 51
- Product states, 59
- Programming languages
 - about hierarchy of abstractions, 367, 380
 - about programming, 364, 377
 - compilers, *see* Compilers, *see* Compilers
 - FORTRAN, 364, 377
 - Haskell, 372, 386
 - Quipper as embedded DSL, 372, 386
 - Quipper oracle construction, 373, 387
 - Silq as embedded DSL, 374, 388
 - Silq oracle construction, 374, 388
 - PSI probabilistic, 374, 388
 - Q language C++class library, 372, 386
 - Q#, 375, 389
 - Silq comparison, 374, 388
 - QASM tool, 367, 380
 - addition via QFT circuit, 267, 272
 - QCL, 368–370, 381–383
 - Quipper comparison, 373, 387
 - Quipper, 372, 386
 - oracle construction, 373, 387
 - proto-Quipper follow-ups, 373, 388
 - QCL comparison, 373, 387
 - Silq comparison, 374, 388
 - resources for information, 375, 389
 - Scaffold, 370, 383
 - classical and quantum constructs, 376, 391
 - entanglement analysis, 371, 385
 - hierarchical QASM, 371, 384
 - transpiler, 370, 384
 - Silq, 374, 388
 - code snippet showcasing, 374, 389
 - oracle construction, 374, 388
- Projection operators, 46
 - 2-dimensional index into matrix, 46
 - controller and controlled qubits not adjacent, 49
 - Hermitian, 46
 - not unitary or reversible, 46
 - outer product representation, 46
 - projective measurements and, 70
- Projection operators (projectors)
 - decomposition with, 119
 - n - it projector construction, 47
- Projective measurements, 69–72
 - about, 70
- ProjectQ commercial system, 375, 389
 - simulator, 383, 397
- PSI probabilistic programming language, 374, 388
- Pure states
 - compiler optimization, 378, 392
 - trace of density matrix, 28
- Python
 - @ operator for matrix multiplication, 13
 - conjugate function, 2
 - about, 11
 - about numpy, 11
 - C++
 - accelerated gate application, 95–101
 - execution speed, 95, 100
 - extending Python with, 96
 - sparse representation, 101
 - sparse representation benchmarked, 103
 - complex numbers, 2
 - operator application, 30–31
 - Tensor class, 11–14
 - * operator for Kronecker product, 13
 - comparing to values, 14
- Q language C++class library, 372, 386
- Q# commercial system (Microsoft), 375, 389
 - programming language, 375, 389
 - Silq comparison, 374, 388
 - Quantum Developer Kit, 375, 389
- QAOA (quantum approximate optimization algorithm), 312, 322
- QASM tool, 367, 380
 - addition via QFT circuit, 267, 272
 - cQASM, 367, 381
 - hierarchical QASM, 371, 384
 - openQASM, 367, 381
 - transpilation dumper function, 90
- qc (quantum circuit) data structure
 - about abstraction, 80
 - constructor, 80
 - gates, 82–85
 - adjoint, 82
 - applying, 82, 98
 - double-controlled gates, 83
 - fast application, 92–95
 - fast application generalized, 94–95
 - faster application with C++, 95–101
 - multi-controlled gates, 84
 - parameterized gates, 83
 - Swap and controlled Swap gates, 85
 - measurements, 86
 - quantum registers, 81
 - qubits added, 81
 - sparse representation, 101–102
 - benchmarking, 103
 - transpilation extension of

- eager mode, 88
- QCL programming language, 368–370, 381–383
- Quipper comparison, 373, 387
- QFT, *see* Quantum Fourier transform
- Quantum Fourier transform
- qHipster simulator, 382, 397
- Qiskit commercial system (IBM), 375, 389
- ALAP scheduling of gates, 379, 393
- simulators, 383, 398
- QRAM model of quantum computing, 365, 366, 378, 379
 - gate approximation, 382, 396
- qsim simulator (Google), 383, 397
- qsimh simulator (Google), 383, 397
- Quantum advantage, 127–136
- Quantum amplitude amplification (QAA), 218–221
- Quantum Amplitude Estimation (QAE), 224–228
- Quantum approximate optimization
 - algorithm (QAOA), 312, 322
- Quantum arithmetic for full adder, 265–271, 276
- Quantum circuit (qc) data structure
 - about abstraction, 80
 - constructor, 80
 - gates, 82–85
 - adjoint, 82
 - applying, 82, 98
 - double-controlled gates, 83
 - fast application, 92–95
 - fast application generalized, 94–95
 - faster application with C⁺⁺, 95–101
 - multi-controlled gates, 84
 - parameterized gates, 83
 - Swap and controlled Swap gates, 85
 - measurements, 86
 - quantum registers, 81
 - qubits added, 81
 - sparse representation, 101–102
 - benchmarking, 103
 - transpilation extension of
 - eager mode, 88
- Quantum circuit model, *see also* Circuits
- Quantum circuit notation, 53–55
 - controlled gates
 - controlled *X* gates, 54
 - controlled *Z* gates, 54
 - Controlled-by-0 Not gate, 54
 - more than one qubit controlling, 54
 - entangler circuits, 59–61
 - fan-out circuits, 126
 - full adder, 123
 - information flow double lines, 54
 - logic circuits, 126
 - measurement, 54
 - oracle for Bernstein–Vazirani algorithm, 164
 - quantum computation, 67
 - qubit order, 53
 - single-qubit operator applied, 53
 - state change depiction, 53
 - state initialization, 53
 - swap test, 150
 - X* gates, 54
- Quantum computers
 - arithmetic multiplication, 271, 276
 - arithmetic via full adder, 123–126
 - quantum arithmetic, 265–271, 276
 - classical computers controlling, 365, 366, 378, 379
 - classical computers simulated by, 128
 - commercial systems, 375, 389
 - compiler design challenges, 364, 378
 - density matrices for theory of, 106
 - environmental challenges, 350–357, 362–369
 - error correction challenges, 357, 370
 - flow control via controlled gates, 48–50
 - QCL programming language, 369, 382
 - Silq programming language, 374, 388
 - GPU coprocessors, 364, 378
 - logic circuits, 126
- Noisy Intermediate Scale Quantum Computers, 299, 308, 365, 378
- operators as ISA of, 29
- QRAM model, 365, 366, 378, 379
 - gate approximation, 382, 396
- quantum computation, 67
 - λ -calculus to express, 372, 386
- quantum registers, 68, 78–80
- simulation, *see* Simulation
- uncomputation, 66–68
 - QCL programming language, 368, 382
 - Silq programming language, 374, 388
 - transpilation intermediate representation, 88
 - trick for saving result, 68
- Quantum computing fundamentals
 - controlled gates, 48–50
 - controlled Not gates, 48–50
 - controlled Not gates constructor function, 50
 - controlled phase gates, 52
 - controlled Swap gates, 51
 - Swap gates, 51
 - data types, 12
 - abstracting, 12
 - complex data type selection, 12
- Quantum computing fundamentals (*conti.*)

- entanglement, 58–65
- measurement
 - examples, 73
 - implementation, 72
 - projective measurements, 69–72
 - quantum mechanics postulates, 68
- multi-controlled gates, 57
 - controlled–controlled Not gates, 55
 - Sleator–Weinfurter construction, 56
- No-Cloning Theorem, 64
 - error correction challenge, 358, 370
- No-Deleting Theorem, 65
- operators, 29–34
 - `apply()` function, 34
 - application, 30–31
 - base class, 29
 - multiple qubits, 31–33
 - padding operators, 33
 - unitary operators, 29
- quantum circuit notation, 53–55
- qubits, 15–17
 - Bloch sphere, 17–19
 - Bloch sphere coordinates for given state, 37
 - constructing in code, 16
- single-qubit gates, 34–48
 - Bloch sphere coordinates, 37
 - flexible phase gates, 42
 - Hadamard gates, 40–41
 - identity gates, 35
 - Pauli matrices, 35
 - phase gates, 41
 - projection operators, 46
 - rotations, 38
 - square roots of gates, 44–46
 - U_3 gates, 43
- states, 19–28
 - binary interpretation, 23–25
 - qubit ordering, 21–22
 - represented as matrices, 28
 - State constructors, 27
 - State member functions, 25–27
 - tensoring states, 20
- Tensor class, 11–14
- uncomputation, 66–68
- Quantum Developer Kit (QDK), 375, 389
 - simulators, 383, 398
- Quantum dot decoherence time, 350, 362
- Quantum error correction
 - about, 350, 357, 362, 369
 - bit-flip errors, 359–361, 371–373
 - Shor’s 9-qubit code, 362, 374
 - compiler optimization and, 376, 390
 - error correction code memory, 357, 369
 - error syndrome, 359, 371
 - phase-flip errors, 361, 373
 - Shor’s 9-qubit code, 362, 374
- quantum computing challenges, 357, 370
- quantum noise, 350–357, 362–369
- repetition code, 357, 369
 - majority voting, 357, 369
 - No-Cloning Theorem, 358, 370
 - quantum repetition code, 358, 370
- resources for information, 363, 376
- Shor’s 9-qubit error correction code, 362, 374
- Quantum Fourier transform (QFT)
 - about, 244
 - algorithm detail
 - about, 258, 261
 - two-qubit QFT online simulation, 264, 268
 - online simulation, 264, 268
- order finding, 279–292, 300
 - continued fractions, 290, 297
 - controlled modular multiplication, 288, 295
 - experimentation, 290, 298
 - main program, 283–284, 290–292
 - modular addition, 286–288, 293–295
 - support routines, 284–286, 292–293
- π approximation, 256–261
- QCL programming language, 370, 383
- QFT operator, 262, 266
 - inverse, 263, 267
- quantum arithmetic
 - addition, 265–271, 276
 - multiplication, 269, 271, 274, 276
 - testing, 270, 275
- quantum phase estimation, 246–256, 259
- Quantum Hello World algorithm, 137
- Quantum information, *see* Information
- Quantum interference, 162
- Quantum IO Monad, 372, 386
- Quantum machine learning
 - Euclidean distance, 327–331, 338–342
 - quantum algorithms that use, 331, 342
 - principal component analysis, 331–336, 342–347
- Quantum mean estimation, 237–239
- Quantum mechanics
 - Copenhagen interpretation, 59
 - hidden state, 58, 61
 - postulates, 68
- Quantum median estimation, 241–243
- Quantum minimum finding, 239–241
- Quantum noise, 350–357, 362–369
 - about, 354, 366
 - amplitude damping, 355, 367
 - channels, 353, 365
 - bit flip and phase flip, 354, 366
 - depolarization, 354, 366

- compiler optimization and noise reduction, 376, 390
 - error correction, 357–363, 369–376
 - gates imprecise, 356, 368
 - modeling via error injection, 355, 367
 - checking bit-flip error correction, 359, 372
 - gates as quantum noise source, 356, 368
 - phase damping, 356, 368
 - quantum error conditions, 351, 363
 - quantum operations, 352, 364
 - operation element, 353, 365
 - operator-sum representation, 353, 365
 - simulation, 382, 383, 397, 398
- Quantum operations
 - operation element, 353, 365
 - operator-sum representation, 353, 365
 - quantum noise, 352, 364
- Quantum parallelism, 162, 169
- Quantum phase estimation (QPE), 246–256, 259
 - detailed derivation, 248–252, 254
 - Hamiltonian eigenvalues, 299, 308
 - implementation, 253–255, 257
 - phase estimation, 247, 248
 - definition, 247, 248
- Quantum counting, 222
- Quantum programming languages
 - about hierarchy of abstractions, 367, 380
 - about programming, 364, 377
 - compilers, *see* Compilers, *see* Compilers
 - Haskell, 372, 386
 - Quipper as embedded DSL, 372, 386
 - Quipper oracle construction, 373, 387
 - Silq as embedded DSL, 374, 388
 - PSI probabilistic, 374, 388
 - Q language C++ class library, 372, 386
 - Q#, 375, 389
 - Silq comparison, 374, 388
 - QASM tool, 367, 380
 - addition via QFT circuit, 267, 272
 - QCL, 368–370, 381–383
 - Quipper comparison, 373, 387
 - Quipper, 372, 386
 - oracle construction, 373, 387
 - proto-Quipper follow-ups, 373, 388
 - QCL comparison, 373, 387
 - Silq comparison, 374, 388
 - resources for information, 375, 389
 - Scaffold, 370, 383
 - classical and quantum constructs, 376, 391
 - entanglement analysis, 371, 385
 - hierarchical QASM, 371, 384
 - transpiler, 370, 384
 - Silq, 374, 388
 - code snippet showcasing, 374, 389
 - oracle construction, 374, 388
- Quantum random circuits (QRC), 129
 - simulation design, 131
 - simulation evaluation, 135
 - simulation implementation, 133
 - simulation metric, 133
- Quantum random walk
 - 1D walk, 294–296, 302–304
 - 2D walk, 295, 304
 - about, 293, 297, 301, 306
 - classical random walk, 293, 301
 - coin toss, 294, 302
 - walking the walk, 296–298, 304–307
- Quantum registers, 78–80
 - about, 76
 - code to create and initialize, 79
 - libq implementation, 384, 399
 - qc data structure, 81
 - compiler optimization and, 376, 390
 - inverting a register, 90
 - QCL programming language, 368, 381
 - reg class, 78
 - result storage, 68
- Quantum simulation, 299, 308
- Quantum state preparation
 - about, 177
 - data encoding, 177–181
 - amplitude encoding, 178
 - basis encoding, 177
 - Hamiltonian encoding, 180
 - Möttönen’s algorithm, 182–188
 - Solovay–Kitaev algorithm, 188–199
 - two- and three-qubit states, 181
- Quantum supremacy, 127–136
- “Quantum supremacy using a programmable superconducting processor” (Arute et al.), 128
- Quantum teleportation, 138–142
 - error correction trick, 359, 371
- Quantum Turing machines, 128
- Qubits, 15–17
 - about the state of a qubit, 15, 19
 - basis states, 15, 19
 - basis states orthonormal, 15
 - collapsing on measurement, 16, 58, 69
 - communicating state of two with one, 142–145
 - equal superposition of adjacent qubits, 41
 - measurement, 72
 - measurement examples, 73
 - probability amplitudes, 15, 16, 19
 - state class constructors, 27

- Qubits (*conti.*)
 superposition via Hadamard gates, 40
 tensor product combined state, 19
 ancilla qubits, 57
 entanglement, 58
 uncomputation, 66
 binary addressing, 23
 Bloch spheres describing, 17–19
 computing coordinates for given state, 37
 cloning or copying impossible, 64
 error correction challenge, 358, 370
 column vectors of complex numbers, 2, 19
 compiler optimization via recycling, 381, 395
 constructing in code, 16
 data structure, 16
 tensoring states, 20
 deleting impossible, 65
 entanglement, 58–65
 junk qubits, 66
 operator application, 30–31
 applied at index specified, 33
 controller and controlled qubits, 48–55
 multiple operators in sequence, 33
 multiple qubits, 31–33
 nonadjacent controller and controlled qubits, 49
 norm preserving, 29
 notation for qubit index applied to, 33
 projection operators extracting subspace, 70
 quantum computation, 67
 qubit ordering, 94
 order of qubits, 21–22
 operator application, 94
 quantum circuit notation, 53
 phase, 26
 quantum circuit notation, 53
 scaling complexity, 76
 tensors constructing, 16
 nbits property, 23
 code, 19
 state for n qubits, 19
 topological limitations to interactions, 50
 Query complexity, 162
 QuEST (Quantum Exact Simulation Toolkit), 383, 397
 Quipper programming language, 372, 386
 oracle automatic construction, 373, 387
 proto-Quipper follow-ups, 373, 388
 QCL comparison, 373, 387
 Silq comparison, 374, 388
 Quirk online simulations, 264, 268
 QX Simulator, 383, 397
 Random circuits, *see* Quantum random circuits (QRC)
 Random number generator, 137
 coin toss, 294, 302
 random combination of 0 or 1 states, 27
 Random walk
 2D walk, 295, 304
 about, 297, 306
 classical random walk, 293, 301
 coin toss, 294, 302
 quantum random walk
 1D walk, 294–296, 302–304
 about, 293, 301
 walking the walk, 296–298, 304–307
 Reduced density operator
 partial trace derivation, 107
 code, 108
 Quirk qubits on Bloch sphere, 264, 269
 Reg class, 78
 Registers, 78–80
 code to create and initialize, 79
 libq implementation, 384, 399
 qc data structure, 81
 compiler optimization and, 376, 390
 inverting a register, 90
 QCL programming language, 368, 381
 reg class, 78
 result storage, 68
 Relaxed peephole optimization, 379, 394
 Renormalization, 72
 Repetition code, 357, 369
 majority voting, 357, 369
 No-Cloning Theorem, 358, 370
 quantum repetition code, 358, 370
 Resources for information
 computational complexity theory, 128
 logical to physical mapping, 382, 396
 quantum error correction, 363, 376
 quantum programming languages, 375, 389
 Quirk online simulator, 264, 268
 Schrödinger equation, 301, 310
 simulators available, 382, 397
 RevKit for reversible oracles, 383, 397
 R_k gates, 42, 43
 Roots (square roots) of gates, 44–46
 `scipy.sqrtm()` function, 45
 Rosen, N., 58
 Rotation axis, 40
 Rotation operators, 38
 axis of rotation, 40
 constructed via U_3 gates, 44
 controlled rotation gates additive, 244
 discrete phase gates, 42
 error source potential, 42
 Hadamard gates, 40–41

- phase gates, 41
 - quantum counting, 222
 - square roots of, 45
- Rotations for encoding, 179
- Row vectors
 - bras in Dirac notation, 2
 - inner products, 3
- RSA encryption algorithm, 273, 278
- S* gates, *see also* Phase gates
 - square root as *T* gate, 45
- Scaffold programming language, 370, 383
 - classical and quantum constructs, 376, 391
 - entanglement analysis, 371, 385
 - hierarchical QASM, 371, 384
 - transpiler, 370, 384
- Scalability
 - about, 76
 - complexity of scaling up, 76, 365, 378
 - controlled gates, 50
 - gate fast application, 92–95
 - hierarchical QASM, 371, 384
 - Quipper programming language, 373, 387
- Scalar products, *see* Inner products
- Scheduling of gates, 378, 392
- Schmidt decomposition, 111–115
- Schrödinger equation
 - qsim simulator, 383, 397
 - resource for more information, 301, 310
 - time-independent for state evolving, 69
 - derivation, 300–301, 309–310
 - variational principle, 301, 310
- Schrödinger full-state simulations, 77
- Schrödinger–Feynman path histories, 100, 129, 136
 - qsimh simulator, 383, 397
- Schrödinger–Feynman Simulation, 78
- scipy
 - installing, 45
 - `sqrtn()` function, 45
- Shor’s 9-qubit error correction code, 362, 374
- Shor’s integer factorization algorithm, 273–279, 285
 - about, 273, 278
 - about phase estimation, 258, 261
 - classical
 - experimentation, 277, 282
 - factorization, 274, 279
 - greatest common divisor, 274, 279
 - modular arithmetic, 273, 278
 - order finding, 275–277, 280–282
 - order finding quantum algorithm
 - continued fractions, 290, 297
 - controlled modular multiplication, 288, 295
 - experimentation, 290, 298
 - main program, 283–284, 290–292
 - modular addition, 286–288, 293–295
 - support routines, 284–286, 292–293
 - sparse representation benchmarked, 103
- Silq programming language, 374, 388
 - code snippet showcasing, 374, 389
 - oracle construction, 374, 388
- Similarity tests
 - about, 150
 - Hadamard test, 155–160
 - inversion test, 160
- Similarity Transformation, 342, 354
- Simon’s algorithm, 176
- Simon’s generalized algorithm, 176
- Simulation
 - about scalability, 76
 - available simulators, 382, 397
 - circuits, 80–86
 - benchmarking, 103
 - double-controlled gates, 83
 - fast gate application, 92–95
 - faster gate application, 95–98
 - finalization, 98
 - gate application, 82
 - measurement, 86
 - multi-controlled gates, 84
 - optimization 1st act, 99
 - parameterized gates, 83
 - qubits, 81
 - sparse representation, 101–102
 - standard gates, 82
 - Swap and controlled Swap gates, 85
 - complexity, 76, 129
 - intermediate representation, 86–91
 - online simulators, 264, 268
 - open-source simulators, 382, 396
 - parallelization of gates, 379, 393
 - quantum Fourier transform online simulation, 264, 268
 - quantum random circuits
 - Google team, 129, 135
 - metric, 133
 - simulation design, 131
 - simulation evaluation, 135
 - simulation implementation, 133
 - quantum registers, 78–80
 - quantum simulating classical computers, 128
 - quantum simulation, 299, 308
- Single-qubit gates, 34–41
 - about constructing multi-qubit operators, 35
 - about quantum gates, 34
 - Bloch sphere coordinates, 37
 - Hadamard gates, 40–41
 - identity gates, 35
 - applied to multiple qubits, 32

- Single-qubit gates (*conti.*)
 - Pauli matrices, 35
 - phase gates, 41
 - discrete phase gates, 42
 - phase shift or kick gates, 43
 - various gates via, 43
 - projection operators, 46
 - quantum circuit notation, 53
 - reversed by conjugate transpose, 29
 - R_k gates, 43
 - rotation operators, 38
 - Hadamard gates, 40–41
 - square roots of gates, 44–46
 - T gates, 43
 - via phase gates, 43
 - $U_1(\lambda)$ gates, 43
 - U_3 gates, 43
 - X gates, 29, 36
 - applied to multiple qubits, 32
 - Y gates, 36
 - Z gates, 36
- Singular value decomposition, 332, 343
- Sleator–Weinfurter construction, 56
- Solovay–Kitaev (SK) algorithm, 188–199
 - about, 188
 - algorithm, 192–194
 - balanced group commutator, 194–197
 - matrix diagonalization function, 196
 - Bloch sphere angle and axis, 189
 - evaluation, 197
 - pre-computing gates, 191
 - similarity metric trace distance, 191
 - theorem, 188
 - universal gates, 189
 - $SU(2)$ group, 189
- Solovay–Kitaev (SK) theorem, 188
- Sparse representation, 101–102
 - benchmarking, 103
 - libquantum library, 101
 - simulation, 382, 397
- SPEC benchmarks, 129
- Spooky action at a distance, 58, 61
 - quantum teleportation, 138–142
- `sqrtm()` function of `scipy`, 45
- Square roots of gates, 44–46
 - `scipy.sqrtm()` function, 45
- State class
 - constructing qubits in code, 16
 - qubit data structure, 16
 - constructors, 27
 - all 0-states or 1-states, 27
 - `density()` function, 106
 - member functions, 25–27
 - `dumper` function, 26
 - probability and amplitudes, 25
 - Tensor class parent, 19
 - `nbits` property, 23
- States, 19
 - about, 12
 - about bit order, 22
 - binary interpretation, 23–25
 - bit index notation, 94
 - basis states of qubits, *see* Basis states
 - cloning, 65
 - collapsing on measurement, 16, 58
 - Born rule, 69
 - measurement definition, 69
 - renormalization, 72
 - density matrices, 28
 - entanglement, 58–65
 - kets representing state of system, 68
 - state evolving via operators, 69
 - maximally mixed state, *see also*
 - Probability amplitudes
 - operator application, 30–31
 - multiple qubits, 31–33
 - projection operators extracting
 - amplitude, 46
 - quantum circuit notation
 - single-qubit operators applied, 53
 - state change depiction, 53
 - state initialization, 53
 - quantum operations, 352, 364
 - qubit ordering, 21–22
 - qubit states, 15
 - similarity tests
 - about, 150
 - Hadamard test, 155–160
 - inversion test, 160
 - swap test, 150–154
 - swap test code, 153
 - swap test for multi-qubit states, 154
 - single-qubit 0 and 1 state constants, 28
 - state preparation
 - about, 177
 - data encoding, 177–181
 - Möttönen’s algorithm, 182–188
 - Solovay–Kitaev algorithm, 188–199
 - two- and three-qubit states, 181
 - state purification technique, 115
 - system state as tensor product, 78
 - tensors constructing qubits, 16
 - code, 19
 - qubit data structure, 16
 - state for n qubits, 19
 - tensor product combined state, 20
- vectors
 - binary interpretation, 23–25
 - column vectors of complex numbers, 2, 16
 - kets representing state of system, 68
 - normalization, 16, 26
 - normalized vectors, 4
 - operator application, 30–31, 69
 - scaling complexity, 76

- unitary operators as norm preserving, 29
- SU(2) group, 189
- Subset sum algorithm, 322–326, 333–337
 - about, 322, 333
 - experiments, 324–326, 335–337
 - implementation, 323, 334
- Subtraction
 - decrement operator, 295, 303
 - testing quantum arithmetic, 270, 275
- Summit supercomputer simulating quantum random circuits, 135
- Superdense coding, 142–145
- Superposition
 - about, 40
 - about measurement, 69
 - error correction challenge, 358, 370
 - about qubits, 15
 - Hadamard gates on qubits, 40
 - equal superposition of adjacent qubits, 41
 - linear combination of basis states, 15
 - maximally mixed state, 111
 - state after operator applied, 53
- SVD, 332, 343
- Swap gates, 51
 - compiler optimization, 379, 394
 - controlled Swap gates, 51
 - qc data structure, 85
- Swap test, 150–154
 - code, 153
 - multi-qubit states, 154
- Sycamore processor, 129
- T* gates
 - square root of *S* gates, 45
 - universal gates, 189
 - via phase gates, 43
- Teleportation, 138–142
 - entanglement teleportation, 145
 - error correction trick, 359, 371
- Tensor class, 11–14
 - * operator for Kronecker product, 13
 - checking if Hermitian or unitary, 14
 - comparing to values, 14
 - is_close() function, 14
 - inner products, 6
 - instantiating, 12
 - ndarray data structure, 12
 - tensor_type() abstraction, 12
 - Kronecker product member function, 13
 - * operator for, 13
 - operators derived from, 29
 - qubit states code, 19, 20
 - State class derived from, 19
 - nbits property, 23
- Tensor products, 5
 - ⊗ operator, 5
 - * operator, 19
 - binary interpretation, 23
 - distributive, 6
 - Kronecker product as, 5, 13
 - mixed-product property, 6
 - multiplication with scalar, 5
 - operators applied to multiple qubits, 31–33
 - multiple operators in sequence, 33
 - product states, 59
 - state of two or more qubits, 19
 - trace of a matrix, 9
- Testing
 - debugging, 14
 - quantum arithmetic, 270, 275
 - tracing out state of one qubit, 109
- Time-evolution encoding, 181
- Toffoli gates, 55
 - logic circuits from, 126
 - multi-controlled *X* gates, 57
 - Sleator–Weinfurter construction, 56
- Tools and techniques
 - density matrices, 106
 - maximal entanglement, 111
 - Pauli representation of operators, 117–120
 - reduced density operators, 107–110
 - Schmidt decomposition, 111–115
 - spectral theorem for normal matrices, 104–106
 - state purification, 115
 - XYX* decomposition, 122–123
 - YZY* decomposition, 120–122
- Trace distance, 191
- Trace of a matrix, 9
 - tensor product, 9
 - trace of outer product of two kets, 9
- Transpilation
 - intermediate representation
 - circuit capabilities of, 88
 - dumper function, 90
 - inverting a register, 90
 - IR base class, 87
 - IR nodes, 86
 - subcircuit control, 89
 - uncomputation, 88
 - Scaffold transpiler, 370, 384
- Transposition
 - involutivity, 2
 - matrix, 2
- Two-qubit quantum Fourier transform online simulator, 264, 268
- $U_1(\lambda)$ gates, 43
 - controlled $U_1(\lambda)$ gate for quantum arithmetic, 266, 270
- U_3 gates, 43

- Uncomputation, 66–68
 - QCL programming language, 368, 382
 - Silq programming language, 374, 388
 - transpilation intermediate representation, 88
 - trick for saving result, 68
- Underscore in function names, 14
- Unitary matrices
 - about, 7
 - checking if Tensor is unitary, 14
 - norm preserving, 7, 29
 - tensoring together with $\otimes$, 13
- Unitary operators, *see also* Gates; Operators
 - invertable, 29
- Universal gates
 - definition, 55, 189
 - QRAM model of quantum computing, 366, 379
 - sets of gates in quantum computing, 51, 55
 - Solovay–Kitaev theorem, 188
 - SU(2) group, 189
- V gates as square root of X gates, 44, 56
- `val2bits()` for decimal to binary, 23
- Variational quantum eigensolver (VQE), 299–312, 322
 - about, 299, 308
 - algorithm, 305–308, 314–317
 - expectation values, 303, 312
 - Hamiltonian type, 302, 311
 - measurement in Pauli bases, 302–305, 312–314
 - measuring eigenvalues, 308–309, 317–319
 - multiple qubits, 310–312, 320–322
 - quantum phase estimation, 299, 308
 - Schrödinger equation, 300–301, 309–310
 - variational principle, 301, 310
 - VQE by peek-a-boo, 320, 331
- Vector database, 331, 342
- Vectors
 - adjoint of, 2
 - binary interpretation, 23–25
 - Bloch vector, 18
 - rotation operators, 38
 - complex numbers, 1
 - dual vectors for a ket, 2
 - eigenvalues, 6
 - eigenvectors, 6
 - Euclidean distance, 327–331, 338–342
 - quantum algorithms that use, 331, 342
 - inner products, 3
 - linear independent, 4
 - norm of, 4
 - normalization, 4
 - orthogonal, 4
 - states
 - basis states of qubits, 15
 - initializing with normalized vector, 28
 - kets representing state of system, 68
 - normalized vectors, 4
 - operator application, 30–31, 69
 - scaling complexity, 76
 - unitary operators as norm preserving, 29
 - tensor products, 5
 - unitary matrices as norm preserving, 7, 29
 - vector spaces, 4
- VQE, *see* Variational quantum eigensolver, *see* Variational quantum eigensolver
- W state entanglement, 63
- Weighted maximum cut, 315, 325
- Wilczek, F., ix, xii
- Wire optimization, 381, 395
- X gates, 29, 36
 - applied to multiple qubits, 32, 92
 - constructed with U_3 gate, 43
 - controlled–controlled X gates, 55
 - Sleator–Weinfurter construction, 56
 - logic circuits from, 126
 - multi-controlled X gates, 57
 - Not gate, 36
 - quantum circuit notation, 54
 - controlled X gates, 54
 - Controlled-by-0 Not gate built, 54
 - square root of as V gate, 44, 56
- XYX decomposition, 122–123
- Y gates, 36
 - square root of, 45
 - yroot gates, 45
- Z gates, 36
 - controlled Z gates, 52
 - phase-flip gates, 36
 - quantum circuit notation, 54
 - controlled Z gates, 54
 - via phase gates, 43
- Z90 gates, *see also* Phase gates
- Zeilinger, Anton, 62
- ZZY decomposition, 120–122