Quantum Computation and Quantum Information

Michael A. Nielsen & Isaac L. Chuang

CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK www.cup.cam.ac.uk
40 West 20th Street, New York, NY 10011-4211, USA www.cup.org
10 Stamford Road, Oakleigh, Melbourne 3166, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain

© Cambridge University Press 2000

This book is in copyright. Subject to statutory exception
and to the provisions of relevant collective licensing agreements,
no reproduction of any part may take place without
the written permission of Cambridge University Press.

First published 2000
Printed in the United Kingdom at the University Press, Cambridge

Typeface Monotype Ehrhardt $10\frac{1}{2}$/13pt *System* LaTeX 2_{ε} [EPC]

A catalogue record of this book is available from the British Library

Library of Congress Cataloguing in Publication data
Nielsen, Michael A., and Chuang, Isaac L.
Quantum Computation and Quantum Information / Michael A. Nielsen and Isaac L. Chuang.
p. cm.
Includes bibliographical references and index.
ISBN 0-521-63503-9
1. Physics. I. Title.
QA401.G47 2000
511′.8–dc21 98-22029 CIP

ISBN 0 521 63235 8 hardback
ISBN 0 521 63503 9 paperback

Contents

Preface *page* xv
Acknowledgements xxi
Nomenclature and notation xxiii

Part I Fundamental concepts 1

1 Introduction and overview 1
- 1.1 Global perspectives 1
  - 1.1.1 History of quantum computation and quantum information 2
  - 1.1.2 Future directions 12
- 1.2 Quantum bits 13
  - 1.2.1 Multiple qubits 16
- 1.3 Quantum computation 17
  - 1.3.1 Single qubit gates 17
  - 1.3.2 Multiple qubit gates 20
  - 1.3.3 Measurements in bases other than the computational basis 22
  - 1.3.4 Quantum circuits 22
  - 1.3.5 Qubit copying circuit? 24
  - 1.3.6 Example: Bell states 25
  - 1.3.7 Example: quantum teleportation 26
- 1.4 Quantum algorithms 28
  - 1.4.1 Classical computations on a quantum computer 29
  - 1.4.2 Quantum parallelism 30
  - 1.4.3 Deutsch's algorithm 32
  - 1.4.4 The Deutsch–Jozsa algorithm 34
  - 1.4.5 Quantum algorithms summarized 36
- 1.5 Experimental quantum information processing 42
  - 1.5.1 The Stern–Gerlach experiment 43
  - 1.5.2 Prospects for practical quantum information processing 46
- 1.6 Quantum information 50
  - 1.6.1 Quantum information theory: example problems 52
  - 1.6.2 Quantum information in a wider context 58

2 Introduction to quantum mechanics 60
- 2.1 Linear algebra 61
  - 2.1.1 Bases and linear independence 62
  - 2.1.2 Linear operators and matrices 63

- 2.1.3 The Pauli matrices 65
- 2.1.4 Inner products 65
- 2.1.5 Eigenvectors and eigenvalues 68
- 2.1.6 Adjoints and Hermitian operators 69
- 2.1.7 Tensor products 71
- 2.1.8 Operator functions 75
- 2.1.9 The commutator and anti-commutator 76
- 2.1.10 The polar and singular value decompositions 78
- 2.2 The postulates of quantum mechanics 80
  - 2.2.1 State space 80
  - 2.2.2 Evolution 81
  - 2.2.3 Quantum measurement 84
  - 2.2.4 Distinguishing quantum states 86
  - 2.2.5 Projective measurements 87
  - 2.2.6 POVM measurements 90
  - 2.2.7 Phase 93
  - 2.2.8 Composite systems 93
  - 2.2.9 Quantum mechanics: a global view 96
- 2.3 Application: superdense coding 97
- 2.4 The density operator 98
  - 2.4.1 Ensembles of quantum states 99
  - 2.4.2 General properties of the density operator 101
  - 2.4.3 The reduced density operator 105
- 2.5 The Schmidt decomposition and purifications 109
- 2.6 EPR and the Bell inequality 111

3 Introduction to computer science 120

- 3.1 Models for computation 122
  - 3.1.1 Turing machines 122
  - 3.1.2 Circuits 129
- 3.2 The analysis of computational problems 135
  - 3.2.1 How to quantify computational resources 136
  - 3.2.2 Computational complexity 138
  - 3.2.3 Decision problems and the complexity classes **P** and **NP** 141
  - 3.2.4 A plethora of complexity classes 150
  - 3.2.5 Energy and computation 153
- 3.3 Perspectives on computer science 161

Part II Quantum computation 171

4 Quantum circuits 171

- 4.1 Quantum algorithms 172
- 4.2 Single qubit operations 174
- 4.3 Controlled operations 177
- 4.4 Measurement 185
- 4.5 Universal quantum gates 188

4.5.1 Two-level unitary gates are universal 189
4.5.2 Single qubit and CNOT gates are universal 191
4.5.3 A discrete set of universal operations 194
4.5.4 Approximating arbitrary unitary gates is generically hard 198
4.5.5 Quantum computational complexity 200
4.6 Summary of the quantum circuit model of computation 202
4.7 Simulation of quantum systems 204
4.7.1 Simulation in action 204
4.7.2 The quantum simulation algorithm 206
4.7.3 An illustrative example 209
4.7.4 Perspectives on quantum simulation 211

5 The quantum Fourier transform and its applications 216
5.1 The quantum Fourier transform 217
5.2 Phase estimation 221
5.2.1 Performance and requirements 223
5.3 Applications: order-finding and factoring 226
5.3.1 Application: order-finding 226
5.3.2 Application: factoring 232
5.4 General applications of the quantum Fourier transform 234
5.4.1 Period-finding 236
5.4.2 Discrete logarithms 238
5.4.3 The hidden subgroup problem 240
5.4.4 Other quantum algorithms? 242

6 Quantum search algorithms 248
6.1 The quantum search algorithm 248
6.1.1 The oracle 248
6.1.2 The procedure 250
6.1.3 Geometric visualization 252
6.1.4 Performance 253
6.2 Quantum search as a quantum simulation 255
6.3 Quantum counting 261
6.4 Speeding up the solution of **NP**-complete problems 263
6.5 Quantum search of an unstructured database 265
6.6 Optimality of the search algorithm 269
6.7 Black box algorithm limits 271

7 Quantum computers: physical realization 277
7.1 Guiding principles 277
7.2 Conditions for quantum computation 279
7.2.1 Representation of quantum information 279
7.2.2 Performance of unitary transformations 281
7.2.3 Preparation of fiducial initial states 281
7.2.4 Measurement of output result 282
7.3 Harmonic oscillator quantum computer 283
7.3.1 Physical apparatus 283

7.3.2 The Hamiltonian 284
7.3.3 Quantum computation 286
7.3.4 Drawbacks 286
7.4 Optical photon quantum computer 287
7.4.1 Physical apparatus 287
7.4.2 Quantum computation 290
7.4.3 Drawbacks 296
7.5 Optical cavity quantum electrodynamics 297
7.5.1 Physical apparatus 298
7.5.2 The Hamiltonian 300
7.5.3 Single-photon single-atom absorption and refraction 303
7.5.4 Quantum computation 306
7.6 Ion traps 309
7.6.1 Physical apparatus 309
7.6.2 The Hamiltonian 317
7.6.3 Quantum computation 319
7.6.4 Experiment 321
7.7 Nuclear magnetic resonance 324
7.7.1 Physical apparatus 325
7.7.2 The Hamiltonian 326
7.7.3 Quantum computation 331
7.7.4 Experiment 336
7.8 Other implementation schemes 343

Part III Quantum information 353

8 Quantum noise and quantum operations 353
8.1 Classical noise and Markov processes 354
8.2 Quantum operations 356
8.2.1 Overview 356
8.2.2 Environments and quantum operations 357
8.2.3 Operator-sum representation 360
8.2.4 Axiomatic approach to quantum operations 366
8.3 Examples of quantum noise and quantum operations 373
8.3.1 Trace and partial trace 374
8.3.2 Geometric picture of single qubit quantum operations 374
8.3.3 Bit flip and phase flip channels 376
8.3.4 Depolarizing channel 378
8.3.5 Amplitude damping 380
8.3.6 Phase damping 383
8.4 Applications of quantum operations 386
8.4.1 Master equations 386
8.4.2 Quantum process tomography 389
8.5 Limitations of the quantum operations formalism 394

9 Distance measures for quantum information 399
9.1 Distance measures for classical information 399
9.2 How close are two quantum states? 403
  9.2.1 Trace distance 403
  9.2.2 Fidelity 409
  9.2.3 Relationships between distance measures 415
9.3 How well does a quantum channel preserve information? 416

10 Quantum error-correction 425
10.1 Introduction 426
  10.1.1 The three qubit bit flip code 427
  10.1.2 Three qubit phase flip code 430
10.2 The Shor code 432
10.3 Theory of quantum error-correction 435
  10.3.1 Discretization of the errors 438
  10.3.2 Independent error models 441
  10.3.3 Degenerate codes 444
  10.3.4 The quantum Hamming bound 444
10.4 Constructing quantum codes 445
  10.4.1 Classical linear codes 445
  10.4.2 Calderbank–Shor–Steane codes 450
10.5 Stabilizer codes 453
  10.5.1 The stabilizer formalism 454
  10.5.2 Unitary gates and the stabilizer formalism 459
  10.5.3 Measurement in the stabilizer formalism 463
  10.5.4 The Gottesman–Knill theorem 464
  10.5.5 Stabilizer code constructions 464
  10.5.6 Examples 467
  10.5.7 Standard form for a stabilizer code 470
  10.5.8 Quantum circuits for encoding, decoding, and correction 472
10.6 Fault-tolerant quantum computation 474
  10.6.1 Fault-tolerance: the big picture 475
  10.6.2 Fault-tolerant quantum logic 482
  10.6.3 Fault-tolerant measurement 489
  10.6.4 Elements of resilient quantum computation 493

11 Entropy and information 500
11.1 Shannon entropy 500
11.2 Basic properties of entropy 502
  11.2.1 The binary entropy 502
  11.2.2 The relative entropy 504
  11.2.3 Conditional entropy and mutual information 505
  11.2.4 The data processing inequality 509
11.3 Von Neumann entropy 510
  11.3.1 Quantum relative entropy 511
  11.3.2 Basic properties of entropy 513
  11.3.3 Measurements and entropy 514

11.3.4 Subadditivity 515
11.3.5 Concavity of the entropy 516
11.3.6 The entropy of a mixture of quantum states 518
11.4 Strong subadditivity 519
11.4.1 Proof of strong subadditivity 519
11.4.2 Strong subadditivity: elementary applications 522

12 Quantum information theory 528
12.1 Distinguishing quantum states and the accessible information 529
12.1.1 The Holevo bound 531
12.1.2 Example applications of the Holevo bound 534
12.2 Data compression 536
12.2.1 Shannon's noiseless channel coding theorem 537
12.2.2 Schumacher's quantum noiseless channel coding theorem 542
12.3 Classical information over noisy quantum channels 546
12.3.1 Communication over noisy classical channels 548
12.3.2 Communication over noisy quantum channels 554
12.4 Quantum information over noisy quantum channels 561
12.4.1 Entropy exchange and the quantum Fano inequality 561
12.4.2 The quantum data processing inequality 564
12.4.3 Quantum Singleton bound 568
12.4.4 Quantum error-correction, refrigeration and Maxwell's demon 569
12.5 Entanglement as a physical resource 571
12.5.1 Transforming bi-partite pure state entanglement 573
12.5.2 Entanglement distillation and dilution 578
12.5.3 Entanglement distillation and quantum error-correction 580
12.6 Quantum cryptography 582
12.6.1 Private key cryptography 582
12.6.2 Privacy amplification and information reconciliation 584
12.6.3 Quantum key distribution 586
12.6.4 Privacy and coherent information 592
12.6.5 The security of quantum key distribution 593

Appendices 608

Appendix 1: Notes on basic probability theory 608

Appendix 2: Group theory 610
A2.1 Basic definitions 610
A2.1.1 Generators 611
A2.1.2 Cyclic groups 611
A2.1.3 Cosets 612
A2.2 Representations 612
A2.2.1 Equivalence and reducibility 612
A2.2.2 Orthogonality 613
A2.2.3 The regular representation 614

A2.3 Fourier transforms 615

Appendix 3: The Solovay–Kitaev theorem 617

Appendix 4: Number theory 625

A4.1 Fundamentals 625
A4.2 Modular arithmetic and Euclid's algorithm 626
A4.3 Reduction of factoring to order-finding 633
A4.4 Continued fractions 635

Appendix 5: Public key cryptography and the RSA cryptosystem 640

Appendix 6: Proof of Lieb's theorem 645

Bibliography 649

Index 665

I Fundamental concepts

1 Introduction and overview

Science offers the boldest metaphysics of the age. It is a thoroughly human construct, driven by the faith that if we dream, press to discover, explain, and dream again, thereby plunging repeatedly into new terrain, the world will somehow come clearer and we will grasp the true strangeness of the universe. And the strangeness will all prove to be connected, and make sense.
– Edward O. Wilson

Information is physical.
– Rolf Landauer

What are the fundamental concepts of quantum computation and quantum information? How did these concepts develop? To what uses may they be put? How will they be presented in this book? The purpose of this introductory chapter is to answer these questions by developing in broad brushstrokes a picture of the field of quantum computation and quantum information. The intent is to communicate a basic understanding of the central concepts of the field, perspective on how they have been developed, and to help you decide how to approach the rest of the book.

Our story begins in Section 1.1 with an account of the historical context in which quantum computation and quantum information has developed. Each remaining section in the chapter gives a brief introduction to one or more fundamental concepts from the field: quantum bits (Section 1.2), quantum computers, quantum gates and quantum circuits (Section 1.3), quantum algorithms (Section 1.4), experimental quantum information processing (Section 1.5), and quantum information and communication (Section 1.6).

Along the way, illustrative and easily accessible developments such as quantum teleportation and some simple quantum algorithms are given, using the basic mathematics taught in this chapter. The presentation is self-contained, and designed to be accessible even without a background in computer science or physics. As we move along, we give pointers to more in-depth discussions in later chapters, where references and suggestions for further reading may also be found.

If as you read you're finding the going rough, skip on to a spot where you feel more comfortable. At points we haven't been able to avoid using a little technical lingo which won't be completely explained until later in the book. Simply accept it for now, and come back later when you understand all the terminology in more detail. The emphasis in this first chapter is on the big picture, with the details to be filled in later.

1.1 Global perspectives

Quantum computation and quantum information is the study of the information processing tasks that can be accomplished using quantum mechanical systems. Sounds pretty simple and obvious, doesn't it? Like many simple but profound ideas it was a long time before anybody thought of doing information processing using quantum mechanical systems. To see why this is the case, we must go back in time and look in turn at each of the fields which have contributed fundamental ideas to quantum computation and quantum information – quantum mechanics, computer science, information theory, and cryptography. As we take our short historical tour of these fields, think of yourself first as a physicist, then as a computer scientist, then as an information theorist, and finally as a cryptographer, in order to get some feel for the disparate perspectives which have come together in quantum computation and quantum information.

1.1.1 History of quantum computation and quantum information

Our story begins at the turn of the twentieth century when a unheralded revolution was underway in science. A series of crises had arisen in physics. The problem was that the theories of physics at that time (now dubbed *classical physics*) were predicting absurdities such as the existence of an 'ultraviolet catastrophe' involving infinite energies, or electrons spiraling inexorably into the atomic nucleus. At first such problems were resolved with the addition of *ad hoc* hypotheses to classical physics, but as a better understanding of atoms and radiation was gained these attempted explanations became more and more convoluted. The crisis came to a head in the early 1920s after a quarter century of turmoil, and resulted in the creation of the modern theory of *quantum mechanics*. Quantum mechanics has been an indispensable part of science ever since, and has been applied with enormous success to everything under and inside the Sun, including the structure of the atom, nuclear fusion in stars, superconductors, the structure of DNA, and the elementary particles of Nature.

What is quantum mechanics? Quantum mechanics is a mathematical framework or set of rules for the construction of physical theories. For example, there is a physical theory known as *quantum electrodynamics* which describes with fantastic accuracy the interaction of atoms and light. Quantum electrodynamics is built up within the framework of quantum mechanics, but it contains specific rules not determined by quantum mechanics. The relationship of quantum mechanics to specific physical theories like quantum electrodynamics is rather like the relationship of a computer's operating system to specific applications software – the operating system sets certain basic parameters and modes of operation, but leaves open how specific tasks are accomplished by the applications.

The rules of quantum mechanics are simple but even experts find them counter-intuitive, and the earliest antecedents of quantum computation and quantum information may be found in the long-standing desire of physicists to better understand quantum mechanics. The best known critic of quantum mechanics, Albert Einstein, went to his grave unreconciled with the theory he helped invent. Generations of physicists since have wrestled with quantum mechanics in an effort to make its predictions more palatable. One of the goals of quantum computation and quantum information is to develop tools which sharpen our intuition about quantum mechanics, and make its predictions more transparent to human minds.

For example, in the early 1980s, interest arose in whether it might be possible to use quantum effects to signal faster than light – a big no-no according to Einstein's theory of relativity. The resolution of this problem turns out to hinge on whether it is possible to *clone* an unknown quantum state, that is, construct a copy of a quantum state. If cloning were possible, then it would be possible to signal faster than light using quantum effects.

However, cloning – so easy to accomplish with classical information (consider the words in front of you, and where they came from!) – turns out not to be possible in general in quantum mechanics. This *no-cloning theorem*, discovered in the early 1980s, is one of the earliest results of quantum computation and quantum information. Many refinements of the no-cloning theorem have since been developed, and we now have conceptual tools which allow us to understand how well a (necessarily imperfect) quantum cloning device might work. These tools, in turn, have been applied to understand other aspects of quantum mechanics.

A related historical strand contributing to the development of quantum computation and quantum information is the interest, dating to the 1970s, of obtaining *complete control over single quantum systems*. Applications of quantum mechanics prior to the 1970s typically involved a gross level of control over a bulk sample containing an enormous number of quantum mechanical systems, none of them directly accessible. For example, superconductivity has a superb quantum mechanical explanation. However, because a superconductor involves a huge (compared to the atomic scale) sample of conducting metal, we can only probe a few aspects of its quantum mechanical nature, with the individual quantum systems constituting the superconductor remaining inaccessible. Systems such as particle accelerators do allow limited access to individual quantum systems, but again provide little control over the constituent systems.

Since the 1970s many techniques for controlling single quantum systems have been developed. For example, methods have been developed for trapping a single atom in an 'atom trap', isolating it from the rest of the world and allowing us to probe many different aspects of its behavior with incredible precision. The scanning tunneling microscope has been used to move single atoms around, creating designer arrays of atoms at will. Electronic devices whose operation involves the transfer of only single electrons have been demonstrated.

Why all this effort to attain complete control over single quantum systems? Setting aside the many technological reasons and concentrating on pure science, the principal answer is that researchers have done this on a hunch. Often the most profound insights in science come when we develop a method for probing a new regime of Nature. For example, the invention of radio astronomy in the 1930s and 1940s led to a spectacular sequence of discoveries, including the galactic core of the Milky Way galaxy, pulsars, and quasars. Low temperature physics has achieved its amazing successes by finding ways to lower the temperatures of different systems. In a similar way, by obtaining complete control over single quantum systems, we are exploring untouched regimes of Nature in the hope of discovering new and unexpected phenomena. We are just now taking our first steps along these lines, and already a few interesting surprises have been discovered in this regime. What else shall we discover as we obtain more complete control over single quantum systems, and extend it to more complex systems?

Quantum computation and quantum information fit naturally into this program. They provide a useful series of challenges at varied levels of difficulty for people devising methods to better manipulate single quantum systems, and stimulate the development of new experimental techniques and provide guidance as to the most interesting directions in which to take experiment. Conversely, the ability to control single quantum systems is essential if we are to harness the power of quantum mechanics for applications to quantum computation and quantum information.

Despite this intense interest, efforts to build quantum information processing systems have resulted in modest success to date. Small quantum computers, capable of doing dozens of operations on a few qubits represent the state of the art in quantum computation. Experimental prototypes for doing *quantum cryptography* – a way of communicating in secret across long distances – have been demonstrated, and are even at the level where they may be useful for some real-world applications. However, it remains a great challenge to physicists and engineers of the future to develop techniques for making large-scale quantum information processing a reality.

Let us turn our attention from quantum mechanics to another of the great intellectual triumphs of the twentieth century, computer science. The origins of computer science are lost in the depths of history. For example, cuneiform tablets indicate that by the time of Hammurabi (circa 1750 B.C.) the Babylonians had developed some fairly sophisticated algorithmic ideas, and it is likely that many of those ideas date to even earlier times.

The modern incarnation of computer science was announced by the great mathematician Alan Turing in a remarkable 1936 paper. Turing developed in detail an abstract notion of what we would now call a programmable computer, a model for computation now known as the *Turing machine*, in his honor. Turing showed that there is a *Universal Turing Machine* that can be used to simulate any other Turing machine. Furthermore, he claimed that the Universal Turing Machine *completely captures* what it means to perform a task by algorithmic means. That is, if an algorithm can be performed on *any* piece of hardware (say, a modern personal computer), then there is an equivalent algorithm for a Universal Turing Machine which performs exactly the same task as the algorithm running on the personal computer. This assertion, known as the *Church–Turing thesis* in honor of Turing and another pioneer of computer science, Alonzo Church, asserts the equivalence between the physical concept of what class of algorithms can be performed on *some physical device* with the rigorous mathematical concept of a Universal Turing Machine. The broad acceptance of this thesis laid the foundation for the development of a rich theory of computer science.

Not long after Turing's paper, the first computers constructed from electronic components were developed. John von Neumann developed a simple theoretical model for how to put together in a practical fashion all the components necessary for a computer to be fully as capable as a Universal Turing Machine. Hardware development truly took off, though, in 1947, when John Bardeen, Walter Brattain, and Will Shockley developed the transistor. Computer hardware has grown in power at an amazing pace ever since, so much so that the growth was codified by Gordon Moore in 1965 in what has come to be known as *Moore's law*, which states that computer power will double for constant cost roughly once every two years.

Amazingly enough, Moore's law has approximately held true in the decades since the 1960s. Nevertheless, most observers expect that this dream run will end some time during the first two decades of the twenty-first century. Conventional approaches to the fabrication of computer technology are beginning to run up against fundamental difficulties of size. Quantum effects are beginning to interfere in the functioning of electronic devices as they are made smaller and smaller.

One possible solution to the problem posed by the eventual failure of Moore's law is to move to a different computing paradigm. One such paradigm is provided by the theory of quantum computation, which is based on the idea of using quantum mechanics to perform computations, instead of classical physics. It turns out that while an ordinary computer can be used to simulate a quantum computer, it appears to be impossible to perform the simulation in an *efficient* fashion. Thus quantum computers offer an essential speed advantage over classical computers. This speed advantage is so significant that many researchers believe that *no* conceivable amount of progress in classical computation would be able to overcome the gap between the power of a classical computer and the power of a quantum computer.

What do we mean by 'efficient' versus 'inefficient' simulations of a quantum computer? Many of the key notions needed to answer this question were actually invented before the notion of a quantum computer had even arisen. In particular, the idea of *efficient* and *inefficient* algorithms was made mathematically precise by the field of *computational complexity*. Roughly speaking, an efficient algorithm is one which runs in time polynomial in the size of the problem solved. In contrast, an inefficient algorithm requires superpolynomial (typically exponential) time. What was noticed in the late 1960s and early 1970s was that it seemed as though the Turing machine model of computation was at least as powerful as any other model of computation, in the sense that a problem which could be solved efficiently in some model of computation could also be solved efficiently in the Turing machine model, by using the Turing machine to simulate the other model of computation. This observation was codified into a strengthened version of the Church–Turing thesis:

> *Any algorithmic process can be simulated efficiently using a Turing machine.*

The key strengthening in the strong Church–Turing thesis is the word *efficiently*. If the strong Church–Turing thesis is correct, then it implies that no matter what type of machine we use to perform our algorithms, that machine can be simulated efficiently using a standard Turing machine. This is an important strengthening, as it implies that for the purposes of analyzing whether a given computational task can be accomplished efficiently, we may restrict ourselves to the analysis of the Turing machine model of computation.

One class of challenges to the strong Church–Turing thesis comes from the field of *analog computation*. In the years since Turing, many different teams of researchers have noticed that certain types of analog computers can efficiently solve problems believed to have no efficient solution on a Turing machine. At first glance these analog computers appear to violate the strong form of the Church–Turing thesis. Unfortunately for analog computation, it turns out that when realistic assumptions about the presence of noise in analog computers are made, their power disappears in all known instances; they cannot efficiently solve problems which are not efficiently solvable on a Turing machine. This lesson – that the effects of realistic noise must be taken into account in evaluating the efficiency of a computational model – was one of the great early challenges of quantum computation and quantum information, a challenge successfully met by the development of a theory of *quantum error-correcting codes* and *fault-tolerant quantum computation*. Thus, unlike analog computation, quantum computation can in principle tolerate a finite amount of noise and still retain its computational advantages.

The first major challenge to the strong Church–Turing thesis arose in the mid 1970s, when Robert Solovay and Volker Strassen showed that it is possible to test whether an integer is prime or composite using a *randomized algorithm*. That is, the Solovay–Strassen test for primality used randomness as an *essential* part of the algorithm. The algorithm did not determine whether a given integer was prime or composite with certainty. Instead, the algorithm could determine that a number was *probably* prime or else composite *with* *certainty*. By repeating the Solovay–Strassen test a few times it is possible to determine with near certainty whether a number is prime or composite. Of especial interest at the time the Solovay–Strassen test was proposed was that no efficient deterministic test for primality was known. Thus, it seemed as though computers with access to a random number generator would be able to efficiently perform computational tasks with no efficient solution on a conventional deterministic Turing machine. This discovery inspired a search for other randomized algorithms which has paid off handsomely, with the field blossoming into a thriving area of research.

Randomized algorithms pose a challenge to the strong Church–Turing thesis, suggesting that there are efficiently soluble problems which, nevertheless, cannot be efficiently solved on a deterministic Turing machine. This challenge appears to be easily resolved by a simple modification of the strong Church–Turing thesis:

Any algorithmic process can be simulated efficiently using a probabilistic Turing machine.

This *ad hoc* modification of the strong Church–Turing thesis should leave you feeling rather queasy. Might it not turn out at some later date that yet another model of computation allows one to efficiently solve problems that are not efficiently soluble within Turing's model of computation? Is there any way we can find a single model of computation which is guaranteed to be able to efficiently simulate any other model of computation?

Motivated by this question, in 1985 David Deutsch asked whether the laws of physics could be use to *derive* an even stronger version of the Church–Turing thesis. Instead of adopting *ad hoc* hypotheses, Deutsch looked to physical theory to provide a foundation for the Church–Turing thesis that would be as secure as the status of that physical theory. In particular, Deutsch attempted to define a computational device that would be capable of efficiently simulating an *arbitrary* physical system. Because the laws of physics are ultimately quantum mechanical, Deutsch was naturally led to consider computing devices based upon the principles of quantum mechanics. These devices, quantum analogues of the machines defined forty-nine years earlier by Turing, led ultimately to the modern conception of a quantum computer used in this book.

At the time of writing it is not clear whether Deutsch's notion of a Universal Quantum Computer is sufficient to efficiently simulate an arbitrary physical system. Proving or refuting this conjecture is one of the great open problems of the field of quantum computation and quantum information. It is possible, for example, that some effect of quantum field theory or an even more esoteric effect based in string theory, quantum gravity or some other physical theory may take us beyond Deutsch's Universal Quantum Computer, giving us a still more powerful model for computation. At this stage, we simply don't know.

What Deutsch's model of a quantum computer did enable was a challenge to the strong form of the Church–Turing thesis. Deutsch asked whether it is possible for a quantum computer to efficiently solve computational problems which have no efficient solution on a classical computer, even a probabilistic Turing machine. He then constructed a simple example suggesting that, indeed, quantum computers might have computational powers exceeding those of classical computers.

This remarkable first step taken by Deutsch was improved in the subsequent decade by many people, culminating in Peter Shor's 1994 demonstration that two enormously important problems – the problem of finding the prime factors of an integer, and the so-called ‘discrete logarithm’ problem – could be solved efficiently on a quantum computer. This attracted widespread interest because these two problems were and still are widely believed to have no efficient solution on a classical computer. Shor’s results are a powerful indication that quantum computers are more powerful than Turing machines, even probabilistic Turing machines. Further evidence for the power of quantum computers came in 1995 when Lov Grover showed that another important problem – the problem of conducting a search through some unstructured search space – could also be sped up on a quantum computer. While Grover’s algorithm did not provide as spectacular a speed-up as Shor’s algorithms, the widespread applicability of search-based methodologies has excited considerable interest in Grover’s algorithm.

At about the same time as Shor’s and Grover’s algorithms were discovered, many people were developing an idea Richard Feynman had suggested in 1982. Feynman had pointed out that there seemed to be essential difficulties in simulating quantum mechanical systems on classical computers, and suggested that building computers based on the principles of quantum mechanics would allow us to avoid those difficulties. In the 1990s several teams of researchers began fleshing this idea out, showing that it is indeed possible to use quantum computers to efficiently simulate systems that have no known efficient simulation on a classical computer. It is likely that one of the major applications of quantum computers in the future will be performing simulations of quantum mechanical systems too difficult to simulate on a classical computer, a problem with profound scientific and technological implications.

What other problems can quantum computers solve more quickly than classical computers? The short answer is that we don’t know. Coming up with good quantum algorithms seems to be *hard*. A pessimist might think that’s because there’s nothing quantum computers are good for other than the applications already discovered! We take a different view. Algorithm design for quantum computers is hard because designers face two difficult problems not faced in the construction of algorithms for classical computers. First, our human intuition is rooted in the classical world. If we use that intuition as an aid to the construction of algorithms, then the algorithmic ideas we come up with will be classical ideas. To design good quantum algorithms one must ‘turn off’ one’s classical intuition for at least part of the design process, using truly quantum effects to achieve the desired algorithmic end. Second, to be truly interesting it is not enough to design an algorithm that is merely quantum mechanical. The algorithm must be *better* than any existing classical algorithm! Thus, it is possible that one may find an algorithm which makes use of truly quantum aspects of quantum mechanics, that is nevertheless not of widespread interest because classical algorithms with comparable performance characteristics exist. The combination of these two problems makes the construction of new quantum algorithms a challenging problem for the future.

Even more broadly, we can ask if there are any generalizations we can make about the power of quantum computers versus classical computers. What is it that makes quantum computers more powerful than classical computers – assuming that this is indeed the case? What class of problems can be solved efficiently on a quantum computer, and how does that class compare to the class of problems that can be solved efficiently on a classical computer? One of the most exciting things about quantum computation and quantum information is how *little* is known about the answers to these questions! It is a great challenge for the future to understand these questions better.

Having come up to the frontier of quantum computation, let’s switch to the history of another strand of thought contributing to quantum computation and quantum information: information theory. At the same time computer science was exploding in the 1940s, another revolution was taking place in our understanding of *communication*. In 1948 Claude Shannon published a remarkable pair of papers laying the foundations for the modern theory of information and communication.

Perhaps the key step taken by Shannon was *to mathematically define the concept of information*. In many mathematical sciences there is considerable flexibility in the choice of fundamental definitions. Try thinking naively for a few minutes about the following question: how would you go about mathematically defining the notion of an information source? Several *different* answers to this problem have found widespread use; however, the definition Shannon came up with seems to be far and away the most fruitful in terms of increased understanding, leading to a plethora of deep results and a theory with a rich structure which seems to accurately reflect many (though not all) real-world communications problems.

Shannon was interested in two key questions related to the communication of information over a communications channel. First, what resources are required to send information over a communications channel? For example, telephone companies need to know how much information they can reliably transmit over a given telephone cable. Second, can information be transmitted in such a way that it is protected against noise in the communications channel?

Shannon answered these two questions by proving the two fundamental theorems of information theory. The first, Shannon's *noiseless channel coding theorem*, quantifies the physical resources required to store the output from an information source. Shannon's second fundamental theorem, the *noisy channel coding theorem*, quantifies how much information it is possible to reliably transmit through a noisy communications channel. To achieve reliable transmission in the presence of noise, Shannon showed that *error-correcting codes* could be used to protect the information being sent. Shannon's noisy channel coding theorem gives an upper limit on the protection afforded by error-correcting codes. Unfortunately, Shannon's theorem does not explicitly give a practically useful set of error-correcting codes to achieve that limit. From the time of Shannon's papers until today, researchers have constructed more and better classes of error-correcting codes in their attempts to come closer to the limit set by Shannon's theorem. A sophisticated theory of error-correcting codes now exists offering the user a plethora of choices in their quest to design a good error-correcting code. Such codes are used in a multitude of places including, for example, compact disc players, computer modems, and satellite communications systems.

Quantum information theory has followed with similar developments. In 1995, Ben Schumacher provided an analogue to Shannon's noiseless coding theorem, and in the process defined the 'quantum bit' or 'qubit' as a tangible physical resource. However, no analogue to Shannon's noisy channel coding theorem is yet known for quantum information. Nevertheless, in analogy to their classical counterparts, a theory of quantum error-correction has been developed which, as already mentioned, allows quantum computers to compute effectively in the presence of noise, and also allows communication over noisy *quantum* channels to take place reliably.

Indeed, classical ideas of error-correction have proved to be enormously important in developing and understanding quantum error-correcting codes. In 1996, two groups working independently, Robert Calderbank and Peter Shor, and Andrew Steane, discovered an important class of quantum codes now known as CSS codes after their initials. This work has since been subsumed by the stabilizer codes, independently discovered by Robert Calderbank, Eric Rains, Peter Shor and Neil Sloane, and by Daniel Gottesman. By building upon the basic ideas of classical linear coding theory, these discoveries greatly facilitated a rapid understanding of quantum error-correcting codes and their application to quantum computation and quantum information.

The theory of quantum error-correcting codes was developed to protect quantum states against noise. What about transmitting ordinary *classical* information using a quantum channel? How efficiently can this be done? A few surprises have been discovered in this arena. In 1992 Charles Bennett and Stephen Wiesner explained how to transmit *two* classical bits of information, while only transmitting *one* quantum bit from sender to receiver, a result dubbed *superdense coding*.

Even more interesting are the results in *distributed quantum computation*. Imagine you have two computers networked, trying to solve a particular problem. How much communication is required to solve the problem? Recently it has been shown that quantum computers can require *exponentially less* communication to solve certain problems than would be required if the networked computers were classical! Unfortunately, as yet these problems are not especially important in a practical setting, and suffer from some undesirable technical restrictions. A major challenge for the future of quantum computation and quantum information is to find problems of real-world importance for which distributed quantum computation offers a substantial advantage over distributed classical computation.

Let's return to information theory proper. The study of information theory begins with the properties of a single communications channel. In applications we often do not deal with a single communications channel, but rather with networks of many channels. The subject of *networked information theory* deals with the information carrying properties of such networks of communications channels, and has been developed into a rich and intricate subject.

By contrast, the study of networked quantum information theory is very much in its infancy. Even for very basic questions we know little about the information carrying abilities of networks of quantum channels. Several rather striking preliminary results have been found in the past few years; however, no unifying theory of networked information theory exists for quantum channels. One example of networked quantum information theory should suffice to convince you of the value such a general theory would have. Imagine that we are attempting to send quantum information from Alice to Bob through a noisy quantum channel. If that channel has zero capacity for quantum information, then it is impossible to reliably send *any* information from Alice to Bob. Imagine instead that we consider two copies of the channel, operating in synchrony. Intuitively it is clear (and can be rigorously justified) that such a channel also has zero capacity to send quantum information. However, if we instead *reverse* the direction of one of the channels, as illustrated in Figure 1.1, it turns out that sometimes we can obtain a non-zero capacity for the transmission of information from Alice to Bob! Counter-intuitive properties like this illustrate the strange nature of quantum information. Better understanding the information carrying properties of networks of quantum channels is a major open problem of quantum computation and quantum information.

Let's switch fields one last time, moving to the venerable old art and science of *cryptography*. Broadly speaking, cryptography is the problem of doing *communication* or

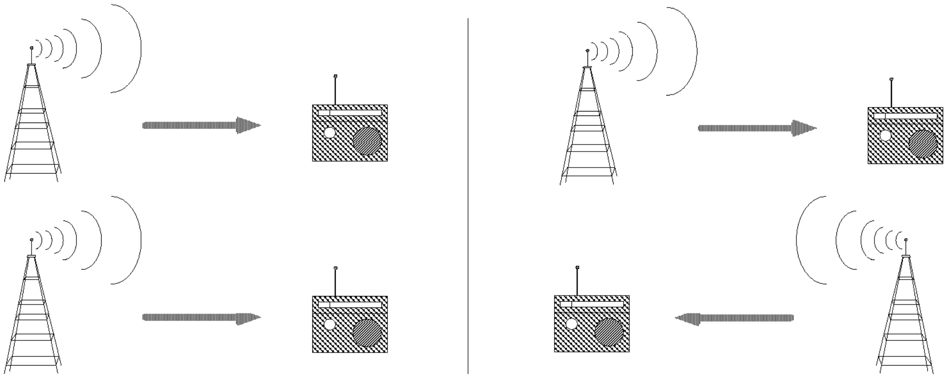

Figure 1.1. Classically, if we have two very noisy channels of zero capacity running side by side, then the combined channel has zero capacity to send information. Not surprisingly, if we reverse the direction of one of the channels, we still have zero capacity to send information. Quantum mechanically, reversing one of the zero capacity channels can actually allow us to send information!

computation involving two or more parties *who may not trust one another*. The best known cryptographic problem is the transmission of secret messages. Suppose two parties wish to communicate in secret. For example, you may wish to give your credit card number to a merchant in exchange for goods, hopefully without any malevolent third party intercepting your credit card number. The way this is done is to use a *cryptographic protocol*. We'll describe in detail how cryptographic protocols work later in the book, but for now it will suffice to make a few simple distinctions. The most important distinction is between *private key cryptosystems* and *public key cryptosystems*.

The way a private key cryptosystem works is that two parties, 'Alice' and 'Bob', wish to communicate by sharing a *private key*, which only they know. The exact form of the key doesn't matter at this point – think of a string of zeroes and ones. The point is that this key is used by Alice to *encrypt* the information she wishes to send to Bob. After Alice encrypts she sends the encrypted information to Bob, who must now recover the original information. Exactly how Alice encrypts the message *depends upon the private key*, so that to recover the original message Bob needs to know the private key, in order to undo the transformation Alice applied.

Unfortunately, private key cryptosystems have some severe problems in many contexts. The most basic problem is how to distribute the keys? In many ways, the key distribution problem is just as difficult as the original problem of communicating in private – a malevolent third party may be eavesdropping on the key distribution, and then use the intercepted key to decrypt some of the message transmission.

One of the earliest discoveries in quantum computation and quantum information was that quantum mechanics can be used to do key distribution in such a way that Alice and Bob's security can not be compromised. This procedure is known as *quantum cryptography* or *quantum key distribution*. The basic idea is to exploit the quantum mechanical principle that observation in general disturbs the system being observed. Thus, if there is an eavesdropper listening in as Alice and Bob attempt to transmit their key, the presence of the eavesdropper will be visible as a disturbance of the communications channel Alice and Bob are using to establish the key. Alice and Bob can then throw out the key bits established while the eavesdropper was listening in, and start over. The first quantum cryptographic ideas were proposed by Stephen Wiesner in the late 1960s, but unfortunately were not accepted for publication! In 1984 Charles Bennett and Gilles Brassard, building on Wiesner's earlier work, proposed a protocol using quantum mechanics to distribute keys between Alice and Bob, without any possibility of a compromise. Since then numerous quantum cryptographic protocols have been proposed, and experimental prototypes developed. At the time of this writing, the experimental prototypes are nearing the stage where they may be useful in limited-scale real-world applications.

The second major type of cryptosystem is the *public key cryptosystem*. Public key cryptosystems don't rely on Alice and Bob sharing a secret key in advance. Instead, Bob simply publishes a 'public key', *which is made available to the general public*. Alice can make use of this public key to encrypt a message which she sends to Bob. What is interesting is that a third party *cannot* use Bob's public key to decrypt the message! Strictly speaking, we shouldn't say *cannot*. Rather, the encryption transformation is chosen in a very clever and non-trivial way so that it is *extremely difficult* (though not impossible) to invert, given only knowledge of the public key. To make inversion easy, Bob has a *secret key* matched to his public key, which together enable him to *easily* perform the decryption. This secret key is not known to anybody other than Bob, who can therefore be confident that only he can read the contents of Alice's transmission, to the extent that it is unlikely that anybody else has the computational power to invert the encryption, given only the public key. Public key cryptosystems solve the key distribution problem by making it unnecessary for Alice and Bob to share a private key before communicating.

Rather remarkably, public key cryptography did not achieve widespread use until the mid-1970s, when it was proposed independently by Whitfield Diffie and Martin Hellman, and by Ralph Merkle, revolutionizing the field of cryptography. A little later, Ronald Rivest, Adi Shamir, and Leonard Adleman developed the *RSA cryptosystem*, which at the time of writing is the most widely deployed public key cryptosystem, believed to offer a fine balance of security and practical usability. In 1997 it was disclosed that these ideas – public key cryptography, the Diffie–Hellman and RSA cryptosystems – were actually invented in the late 1960s and early 1970s by researchers working at the British intelligence agency GCHQ.

The key to the security of public key cryptosystems is that it should be difficult to invert the encryption stage if only the public key is available. For example, it turns out that inverting the encryption stage of RSA is a problem closely related to factoring. Much of the presumed security of RSA comes from the belief that factoring is a problem hard to solve on a classical computer. However, Shor's fast algorithm for factoring on a quantum computer could be used to break RSA! Similarly, there are other public key cryptosystems which can be broken if a fast algorithm for solving the discrete logarithm problem – like Shor's quantum algorithm for discrete logarithm – were known. This practical application of quantum computers to the breaking of cryptographic codes has excited much of the interest in quantum computation and quantum information.

We have been looking at the historical antecedents for quantum computation and quantum information. Of course, as the field has grown and matured, it has sprouted its own subfields of research, whose antecedents lie mainly within quantum computation and quantum information.

Perhaps the most striking of these is the study of *quantum entanglement*. Entanglement is a uniquely quantum mechanical *resource* that plays a key role in many of the most interesting applications of quantum computation and quantum information; entanglement is iron to the classical world's bronze age. In recent years there has been a tremendous effort trying to better understand the properties of entanglement considered as a fundamental resource of Nature, of comparable importance to energy, information, entropy, or any other fundamental resource. Although there is as yet no complete theory of entanglement, some progress has been made in understanding this strange property of quantum mechanics. It is hoped by many researchers that further study of the properties of entanglement will yield insights that facilitate the development of new applications in quantum computation and quantum information.

1.1.2 Future directions

We've looked at some at the history and present status of quantum computation and quantum information. What of the future? What can quantum computation and quantum information offer to science, to technology, and to humanity? What benefits does quantum computation and quantum information confer upon its parent fields of computer science, information theory, and physics? What are the key open problems of quantum computation and quantum information? We will make a few very brief remarks about these overarching questions before moving onto more detailed investigations.

Quantum computation and quantum information has taught us to *think physically about computation*, and we have discovered that this approach yields many new and exciting capabilities for information processing and communication. Computer scientists and information theorists have been gifted with a new and rich paradigm for exploration. Indeed, in the broadest terms we have learned that *any physical theory*, not just quantum mechanics, may be used as the basis for a theory of information processing and communication. The fruits of these explorations may one day result in information processing devices with capabilities far beyond today's computing and communications systems, with concomitant benefits and drawbacks for society as a whole.

Quantum computation and quantum information certainly offer challenges aplenty to physicists, but it is perhaps a little subtle what quantum computation and quantum information offers to physics in the long term. We believe that just as we have learned to think physically about computation, we can also learn to *think computationally about physics*. Whereas physics has traditionally been a discipline focused on understanding 'elementary' objects and simple systems, many interesting aspects of Nature arise only when things become larger and more complicated. Chemistry and engineering deal with such complexity to some extent, but most often in a rather *ad hoc* fashion. One of the messages of quantum computation and information is that new tools are available for traversing the gulf between the small and the relatively complex: computation and algorithms provide systematic means for constructing and understanding such systems. Applying ideas from these fields is already beginning to yield new insights into physics. It is our hope that this perspective will blossom in years to come into a fruitful way of understanding all aspects of physics.

We've briefly examined some of the key motivations and ideas underlying quantum computation and quantum information. Over the remaining sections of this chapter we give a more technical but still accessible introduction to these motivations and ideas, with the hope of giving you a bird's-eye view of the field as it is presently poised.

1.2 Quantum bits

The *bit* is the fundamental concept of classical computation and classical information. Quantum computation and quantum information are built upon an analogous concept, the *quantum bit*, or *qubit* for short. In this section we introduce the properties of single and multiple qubits, comparing and contrasting their properties to those of classical bits.

What is a qubit? We're going to describe qubits as *mathematical objects* with certain specific properties. 'But hang on', you say, 'I thought qubits were physical objects.' It's true that qubits, like bits, are realized as actual physical systems, and in Section 1.5 and Chapter 7 we describe in detail how this connection between the abstract mathematical point of view and real systems is made. However, for the most part we treat qubits as abstract mathematical objects. The beauty of treating qubits as abstract entities is that it gives us the freedom to construct a general theory of quantum computation and quantum information which does not depend upon a specific system for its realization.

What then is a qubit? Just as a classical bit has a *state* – either 0 or 1 – a qubit also has a state. Two possible states for a qubit are the states $|0\rangle$ and $|1\rangle$, which as you might guess correspond to the states 0 and 1 for a classical bit. Notation like '$|\ \rangle$' is called the *Dirac notation*, and we'll be seeing it often, as it's the standard notation for states in quantum mechanics. The difference between bits and qubits is that a qubit can be in a state *other* than $|0\rangle$ or $|1\rangle$. It is also possible to form *linear combinations* of states, often called *superpositions*:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle. \tag{1.1}$$

The numbers α and β are complex numbers, although for many purposes not much is lost by thinking of them as real numbers. Put another way, the state of a qubit is a vector in a two-dimensional complex vector space. The special states $|0\rangle$ and $|1\rangle$ are known as *computational basis states*, and form an orthonormal basis for this vector space.

We can examine a bit to determine whether it is in the state 0 or 1. For example, computers do this all the time when they retrieve the contents of their memory. Rather remarkably, we cannot examine a qubit to determine its quantum state, that is, the values of α and β. Instead, quantum mechanics tells us that we can only acquire much more restricted information about the quantum state. When we measure a qubit we get either the result 0, with probability $|\alpha|^2$, or the result 1, with probability $|\beta|^2$. Naturally, $|\alpha|^2 + |\beta|^2 = 1$, since the probabilities must sum to one. Geometrically, we can interpret this as the condition that the qubit's state be normalized to length 1. Thus, in general a qubit's state is a unit vector in a two-dimensional complex vector space.

This dichotomy between the unobservable state of a qubit and the observations we can make lies at the heart of quantum computation and quantum information. In most of our abstract models of the world, there is a direct correspondence between elements of the abstraction and the real world, just as an architect's plans for a building are in correspondence with the final building. The lack of this direct correspondence in quantum mechanics makes it difficult to intuit the behavior of quantum systems; however, there is an indirect correspondence, for qubit states can be manipulated and transformed in ways which lead to measurement outcomes which depend distinctly on the different properties of the state. Thus, these quantum states have real, experimentally verifiable consequences, which we shall see are essential to the power of quantum computation and quantum information.

The ability of a qubit to be in a superposition state runs counter to our 'common sense' understanding of the physical world around us. A classical bit is like a coin: either heads or tails up. For imperfect coins, there may be intermediate states like having it balanced on an edge, but those can be disregarded in the ideal case. By contrast, a qubit can exist in a *continuum* of states between $|0\rangle$ and $|1\rangle$ – until it is observed. Let us emphasize again that when a qubit is measured, it only ever gives '0' or '1' as the measurement result – probabilistically. For example, a qubit can be in the state

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle , \tag{1.2}$$

which, when measured, gives the result 0 fifty percent ($|1/\sqrt{2}|^2$) of the time, and the result 1 fifty percent of the time. We will return often to this state, which is sometimes denoted $|+\rangle$.

Despite this strangeness, qubits are decidedly real, their existence and behavior extensively validated by experiments (discussed in Section 1.5 and Chapter 7), and many different physical systems can be used to realize qubits. To get a concrete feel for how a qubit can be realized it may be helpful to list some of the ways this realization may occur: as the two different polarizations of a photon; as the alignment of a nuclear spin in a uniform magnetic field; as two states of an electron orbiting a single atom such as shown in Figure 1.2. In the atom model, the electron can exist in either the so-called 'ground' or 'excited' states, which we'll call $|0\rangle$ and $|1\rangle$, respectively. By shining light on the atom, with appropriate energy and for an appropriate length of time, it is possible to move the electron from the $|0\rangle$ state to the $|1\rangle$ state and vice versa. But more interestingly, by reducing the time we shine the light, an electron initially in the state $|0\rangle$ can be moved 'halfway' between $|0\rangle$ and $|1\rangle$, into the $|+\rangle$ state.

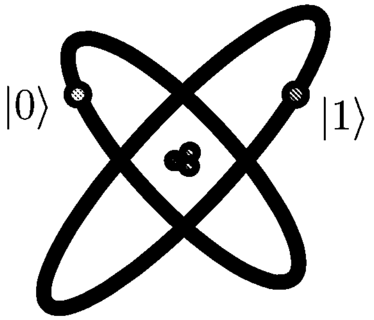

Figure 1.2. Qubit represented by two electronic levels in an atom.

Naturally, a great deal of attention has been given to the 'meaning' or 'interpretation' that might be attached to superposition states, and of the inherently probabilistic nature of observations on quantum systems. However, by and large, we shall not concern ourselves with such discussions in this book. Instead, our intent will be to develop mathematical and conceptual pictures which are predictive.

One picture useful in thinking about qubits is the following geometric representation.

Because $|\alpha|^2 + |\beta|^2 = 1$, we may rewrite Equation (1.1) as

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle\right), \tag{1.3}$$

where θ, φ and γ are real numbers. In Chapter 2 we will see that we can *ignore* the factor of $e^{i\gamma}$ out the front, because it has *no observable effects*, and for that reason we can effectively write

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle. \tag{1.4}$$

The numbers θ and φ define a point on the unit three-dimensional sphere, as shown in Figure 1.3. This sphere is often called the *Bloch sphere*; it provides a useful means of visualizing the state of a single qubit, and often serves as an excellent testbed for ideas about quantum computation and quantum information. Many of the operations on single qubits which we describe later in this chapter are neatly described within the Bloch sphere picture. However, it must be kept in mind that this intuition is limited because there is no simple generalization of the Bloch sphere known for multiple qubits.

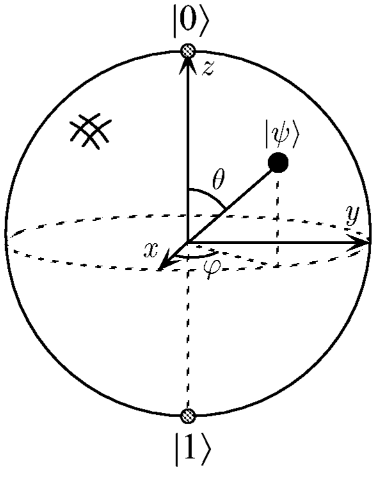

Figure 1.3. Bloch sphere representation of a qubit.

How much information is represented by a qubit? Paradoxically, there are an infinite number of points on the unit sphere, so that in principle one could store an entire text of Shakespeare in the infinite binary expansion of θ. However, this conclusion turns out to be misleading, because of the behavior of a qubit when observed. Recall that measurement of a qubit will give *only* either 0 or 1. Furthermore, measurement *changes* the state of a qubit, collapsing it from its superposition of $|0\rangle$ and $|1\rangle$ to the specific state consistent with the measurement result. For example, if measurement of $|+\rangle$ gives 0, then the post-measurement state of the qubit will be $|0\rangle$. Why does this type of collapse occur? Nobody knows. As discussed in Chapter 2, this behavior is simply one of the *fundamental postulates* of quantum mechanics. What is relevant for our purposes is that from a single measurement one obtains only a single bit of information about the state of the qubit, thus resolving the apparent paradox. It turns out that only if infinitely many identically prepared qubits were measured would one be able to determine α and β for a qubit in the state given in Equation (1.1).

But an even more interesting question to ask might be: how much information is represented by a qubit *if we do not measure it*? This is a trick question, because how can one quantify information if it cannot be measured? Nevertheless, there is something conceptually important here, because when Nature evolves a closed quantum system of qubits, not performing any 'measurements', she apparently does keep track of all the continuous variables describing the state, like α and β. In a sense, in the state of a qubit, Nature conceals a great deal of 'hidden information'. And even more interestingly, we will see shortly that the potential amount of this extra 'information' grows exponentially with the number of qubits. Understanding this hidden *quantum information* is a question that we grapple with for much of this book, and which lies at the heart of what makes quantum mechanics a powerful tool for information processing.

1.2.1 Multiple qubits

Hilbert space is a big place.
– Carlton Caves

Suppose we have two qubits. If these were two classical bits, then there would be four possible states, 00, 01, 10, and 11. Correspondingly, a two qubit system has four *computational basis states* denoted $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. A pair of qubits can also exist in superpositions of these four states, so the quantum state of two qubits involves associating a complex coefficient – sometimes called an *amplitude* – with each computational basis state, such that the state vector describing the two qubits is

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle. \tag{1.5}$$

Similar to the case for a single qubit, the measurement result x ($= 00, 01, 10$ or 11) occurs with probability $|\alpha_x|^2$, with the state of the qubits after the measurement being $|x\rangle$. The condition that probabilities sum to one is therefore expressed by the *normalization* condition that $\sum_{x\in\{0,1\}^2} |\alpha_x|^2 = 1$, where the notation '$\{0,1\}^2$' means 'the set of strings of length two with each letter being either zero or one'. For a two qubit system, we could measure just a subset of the qubits, say the first qubit, and you can probably guess how this works: measuring the first qubit alone gives 0 with probability $|\alpha_{00}|^2 + |\alpha_{01}|^2$, leaving the post-measurement state

$$|\psi'\rangle = \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}. \tag{1.6}$$

Note how the post-measurement state is *re-normalized* by the factor $\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}$ so that it still satisfies the normalization condition, just as we expect for a legitimate quantum state.

An important two qubit state is the *Bell state* or *EPR pair*,

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}. \tag{1.7}$$

This innocuous-looking state is responsible for many surprises in quantum computation and quantum information. It is the key ingredient in quantum teleportation and superdense coding, which we'll come to in Section 1.3.7 and Section 2.3, respectively, and the prototype for many other interesting quantum states. The Bell state has the property that upon measuring the first qubit, one obtains two possible results: 0 with probability 1/2, leaving the post-measurement state $|\varphi'\rangle = |00\rangle$, and 1 with probability 1/2, leaving $|\varphi'\rangle = |11\rangle$. As a result, a measurement of the second qubit always gives the same result as the measurement of the first qubit. That is, the measurement outcomes are *correlated.* Indeed, it turns out that other types of measurements can be performed on the Bell state, by first applying some operations to the first or second qubit, and that interesting correlations still exist between the result of a measurement on the first and second qubit. These correlations have been the subject of intense interest ever since a famous paper by Einstein, Podolsky and Rosen, in which they first pointed out the strange properties of states like the Bell state. EPR's insights were taken up and greatly improved by John Bell, who proved an amazing result: the measurement correlations in the Bell state are *stronger than could ever exist between classical systems.* These results, described in detail in Section 2.6, were the first intimation that quantum mechanics allows information processing beyond what is possible in the classical world.

More generally, we may consider a system of n qubits. The computational basis states of this system are of the form $|x_1x_2 \ldots x_n\rangle$, and so a quantum state of such a system is specified by 2^n amplitudes. For $n = 500$ this number is larger than the estimated number of atoms in the Universe! Trying to store all these complex numbers would not be possible on any conceivable classical computer. Hilbert space is indeed a big place. In principle, however, Nature manipulates such enormous quantities of data, even for systems containing only a few hundred atoms. It is as if Nature were keeping 2^{500} hidden pieces of scratch paper on the side, on which she performs her calculations as the system evolves. This enormous potential computational power is something we would very much like to take advantage of. But how can we think of quantum mechanics as computation?

1.3 Quantum computation

Changes occurring to a quantum state can be described using the language of *quantum computation.* Analogous to the way a classical computer is built from an electrical circuit containing wires and logic gates, a quantum computer is built from a *quantum circuit* containing wires and elementary *quantum gates* to carry around and manipulate the quantum information. In this section we describe some simple quantum gates, and present several example circuits illustrating their application, including a circuit which teleports qubits!

1.3.1 Single qubit gates

Classical computer circuits consist of *wires* and *logic gates.* The wires are used to carry information around the circuit, while the logic gates perform manipulations of the information, converting it from one form to another. Consider, for example, classical single bit logic gates. The only non-trivial member of this class is the NOT gate, whose operation is defined by its *truth table*, in which $0 \to 1$ and $1 \to 0$, that is, the 0 and 1 states are interchanged.

Can an analogous quantum NOT gate for qubits be defined? Imagine that we had some process which took the state $|0\rangle$ to the state $|1\rangle$, and vice versa. Such a process would obviously be a good candidate for a quantum analogue to the NOT gate. However, specifying the action of the gate on the states $|0\rangle$ and $|1\rangle$ does not tell us what happens to superpositions of the states $|0\rangle$ and $|1\rangle$, without further knowledge about the properties of quantum gates. In fact, the quantum NOT gate acts *linearly*, that is, it takes the state

$$\alpha|0\rangle + \beta|1\rangle \tag{1.8}$$

to the corresponding state in which the role of $|0\rangle$ and $|1\rangle$ have been interchanged,

$$\alpha|1\rangle + \beta|0\rangle. \tag{1.9}$$

Why the quantum NOT gate acts linearly and not in some nonlinear fashion is a very interesting question, and the answer is not at all obvious. It turns out that this linear behavior is a general property of quantum mechanics, and very well motivated empirically; moreover, nonlinear behavior can lead to apparent paradoxes such as time travel, faster-than-light communication, and violations of the second laws of thermodynamics. We'll explore this point in more depth in later chapters, but for now we'll just take it as given.

There is a convenient way of representing the quantum NOT gate in matrix form, which follows directly from the linearity of quantum gates. Suppose we define a matrix X to represent the quantum NOT gate as follows:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{1.10}$$

(The notation X for the quantum NOT is used for historical reasons.) If the quantum state $\alpha|0\rangle + \beta|1\rangle$ is written in a vector notation as

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{1.11}$$

with the top entry corresponding to the amplitude for $|0\rangle$ and the bottom entry the amplitude for $|1\rangle$, then the corresponding output from the quantum NOT gate is

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}. \tag{1.12}$$

Notice that the action of the NOT gate is to take the state $|0\rangle$ and replace it by the state corresponding to the first column of the matrix X. Similarly, the state $|1\rangle$ is replaced by the state corresponding to the second column of the matrix X.

So quantum gates on a single qubit can be described by two by two matrices. Are there any constraints on what matrices may be used as quantum gates? It turns out that there are. Recall that the normalization condition requires $|\alpha|^2 + |\beta|^2 = 1$ for a quantum state $\alpha|0\rangle + \beta|1\rangle$. This must also be true of the quantum state $|\psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$ after the gate has acted. It turns out that the appropriate condition on the matrix representing the gate is that the matrix U describing the single qubit gate be *unitary*, that is $U^\dagger U = I$, where $U^\dagger$ is the *adjoint* of U (obtained by transposing and then complex conjugating U), and I is the two by two identity matrix. For example, for the NOT gate it is easy to verify that $X^\dagger X = I$.

Amazingly, this *unitarity* constraint is the *only* constraint on quantum gates. Any unitary matrix specifies a valid quantum gate! The interesting implication is that in contrast to the classical case, where only one non-trivial single bit gate exists – the NOT

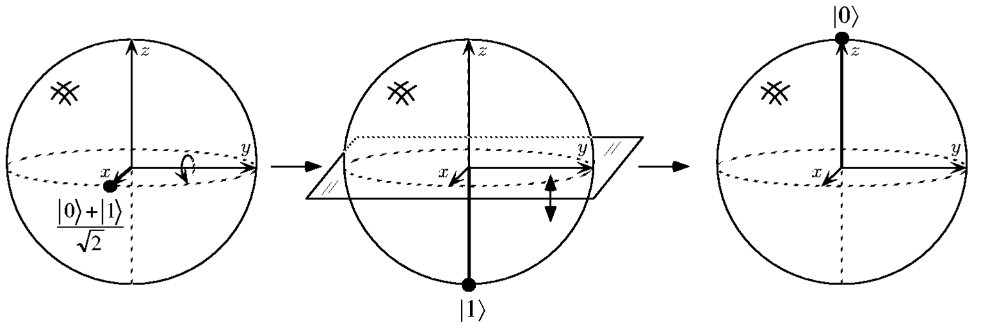

Figure 1.4. Visualization of the Hadamard gate on the Bloch sphere, acting on the input state $(|0\rangle + |1\rangle)/\sqrt{2}$.

gate – there are many non-trivial single qubit gates. Two important ones which we shall use later are the Z gate:

$$Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{1.13}$$

which leaves $|0\rangle$ unchanged, and flips the sign of $|1\rangle$ to give $-|1\rangle$, and the *Hadamard* gate,

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{1.14}$$

This gate is sometimes described as being like a 'square-root of NOT' gate, in that it turns a $|0\rangle$ into $(|0\rangle + |1\rangle)/\sqrt{2}$ (first column of H), 'halfway' between $|0\rangle$ and $|1\rangle$, and turns $|1\rangle$ into $(|0\rangle - |1\rangle)/\sqrt{2}$ (second column of H), which is also 'halfway' between $|0\rangle$ and $|1\rangle$. Note, however, that H^2 is not a NOT gate, as simple algebra shows that $H^2 = I$, and thus applying H twice to a state does nothing to it.

The Hadamard gate is one of the most useful quantum gates, and it is worth trying to visualize its operation by considering the Bloch sphere picture. In this picture, it turns out that single qubit gates correspond to rotations and reflections of the sphere. The Hadamard operation is just a rotation of the sphere about the $\hat{y}$ axis by 90°, followed by a reflection through the $\hat{x}$-$\hat{y}$ plane, as illustrated in Figure 1.4. Some important single qubit gates are shown in Figure 1.5, and contrasted with the classical case.

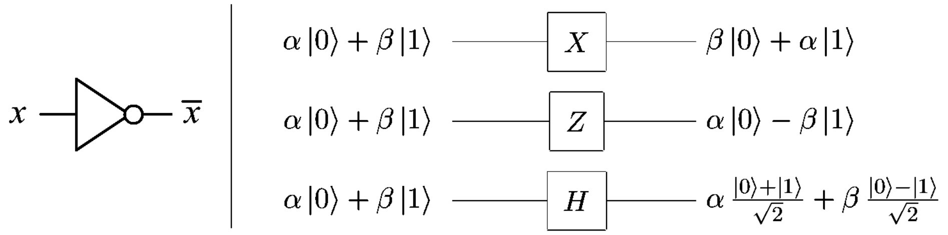
x
$\overline{x}$
$\alpha|0\rangle + \beta|1\rangle$
X
$\beta|0\rangle + \alpha|1\rangle$
$\alpha|0\rangle + \beta|1\rangle$
Z
$\alpha|0\rangle - \beta|1\rangle$
$\alpha|0\rangle + \beta|1\rangle$
H
$\alpha\frac{|0\rangle+|1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle-|1\rangle}{\sqrt{2}}$

Figure 1.5. Single bit (left) and qubit (right) logic gates.

There are infinitely many two by two unitary matrices, and thus infinitely many single qubit gates. However, it turns out that the properties of the complete set can be understood from the properties of a much smaller set. For example, as explained in Box 1.1, an arbitrary single qubit unitary gate can be decomposed as a product of rotations

$$\begin{bmatrix} \cos\frac{\gamma}{2} & -\sin\frac{\gamma}{2} \\ \sin\frac{\gamma}{2} & \cos\frac{\gamma}{2} \end{bmatrix}, \tag{1.15}$$

and a gate which we'll later understand as being a rotation about the $\hat{z}$ axis,

$$\begin{bmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{bmatrix}, \tag{1.16}$$

together with a *(global) phase shift* – a constant multiplier of the form $e^{i\alpha}$. These gates can be broken down further – we don't need to be able to do these gates for arbitrary α, β and γ, but can build arbitrarily good approximations to such gates using only certain special *fixed* values of α, β and γ. In this way it is possible to build up an arbitrary single qubit gate using a *finite* set of quantum gates. More generally, an arbitrary quantum computation on any number of qubits can be generated by a finite set of gates that is said to be *universal* for quantum computation. To obtain such a universal set we first need to introduce some quantum gates involving multiple qubits.

Box 1.1: Decomposing single qubit operations

In Section 4.2 starting on page 174 we prove that an arbitrary 2×2 unitary matrix may be decomposed as

$$U = e^{i\alpha} \begin{bmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{bmatrix} \begin{bmatrix} \cos\frac{\gamma}{2} & -\sin\frac{\gamma}{2} \\ \sin\frac{\gamma}{2} & \cos\frac{\gamma}{2} \end{bmatrix}, \begin{bmatrix} e^{-i\delta/2} & 0 \\ 0 & e^{i\delta/2} \end{bmatrix}, \tag{1.17}$$

where α, β, γ, and δ are real-valued. Notice that the second matrix is just an ordinary rotation. It turns out that the first and last matrices can also be understood as rotations in a different plane. This decomposition can be used to give an exact prescription for performing an *arbitrary* single qubit quantum logic gate.

1.3.2 Multiple qubit gates

Now let us generalize from one to multiple qubits. Figure 1.6 shows five notable multiple bit classical gates, the AND, OR, XOR (exclusive-OR), NAND and NOR gates. An important theoretical result is that any function on bits can be computed from the composition of NAND gates alone, which is thus known as a *universal* gate. By contrast, the XOR alone or even together with NOT is not universal. One way of seeing this is to note that applying an XOR gate does not change the total parity of the bits. As a result, any circuit involving only NOT and XOR gates will, if two inputs x and y have the same parity, give outputs with the same parity, restricting the class of functions which may be computed, and thus precluding universality.

The prototypical multi-qubit quantum logic gate is the *controlled*-NOT or CNOT gate. This gate has two input qubits, known as the *control* qubit and the *target* qubit, respectively. The circuit representation for the CNOT is shown in the top right of Figure 1.6; the top line represents the control qubit, while the bottom line represents the target