The Art of Fluid Animation

Jos Stam

CRC Press
Taylor & Francis Group
AN A K PETERS BOOK

The Art of Fluid Animation

The Art of Fluid Animation

Jos Stam

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
AN A K PETERS BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Version Date: 20150921

International Standard Book Number-13: 978-1-4987-0021-4 (eBook - PDF)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

To Pam and Gillian

This is a small replica of the northern portal of the Urnes Stavkyrkje *in Norway. This church was built in 1130, near the end of the Viking domination of Northern Europe. Vikings were highly skilled artists, not just long-haired brutish warriors. I took this photograph at the British Museum in London, England.*

Contents

Preface, xi

Acknowledgments, xix

Author, xxi

CHAPTER 1 ▪ Introduction 1

CHAPTER 2 ▪ Observations, Equations, and Numbers 11

2.1 BEAVERS, CAVE PERSONS, AND FIRE 11

2.2 FROM CAVES TO GREEKS: ARCHIMEDES, GOLD, AND MEDALS 13

2.3 LONG, CURLY HAIRED MATHEMATICIANS, THE ABYSS AND THE AIRBRUSH 16

CHAPTER 3 ▪ Euler–Newton Equations or Navier–Stokes Equations 35

3.1 LEONARDO DA VINCI 35

3.2 EULER AND CONTINUITY 36

3.3 INCOMPRESSIBILITY, CONTINUITY, HELMHOLTZ, AND HODGE THEORY 44

3.4 EULER AND THE MOTION OF FLUIDS 49

3.5 NEWTON AND VISCOSITY 55

3.6 NAVIER AND STOKES AND THEIR EQUATIONS 64

3.7 BOUNDARIES, BOUNDARIES, BOUNDARIES, BOUNDARIES, AND BOUNDARIES 69

CHAPTER 4 ■ The Early Days of Computational Fluid Dynamics 81

CHAPTER 5 ■ Kolmogorov and Turbulence 91

CHAPTER 6 ■ Introduction to Fluid Animation 97

6.1 DISCRETIZE! BUGS, GRIDS, AND BUGS MOVING THROUGH GRIDS 98

6.1.1 Bugs 99

6.1.2 Grids 103

6.1.3 Bugs Moving through Grids 107

6.1.4 Semi-Lagrangian 108

6.1.5 PIC 112

CHAPTER 7 ■ Intermezzi 117

7.1 INTERMEZZO UNO: LINEAR SYSTEMS 118

7.2 INTERMEZZO DUE: THE GENERAL SOLUTION OF A LINEAR SYSTEM 124

7.3 INTERMEZZO TRE: CIRCULANT MATRICES AND THE FOURIER TRANSFORM 132

7.4 INTERMEZZO QUATTRO: NUMERICAL SOLUTION OF LINEAR SYSTEMS 141

CHAPTER 8 ■ A Simple Fluid Solver 155

8.1 A MATH HORROR FLICK: OPERATOR SPLITTING 159

8.2 CODE PLEASE? 160

8.2.1 Moving Densities 164

8.2.2 Moving Fluids That Change Themselves: Nonlinearity 170

8.3 BUGS CRAWLING ON DONUTS, THE FFT, AND ~60 LINES OF C CODE 176

8.4 FOUR-DIMENSIONAL TURBULENT VECTOR FIELDS AND TURBULENCE 186

8.5 DECORATING FLUIDS 189

CHAPTER 9 ▪ The Little Computers Who Can Handle Fluids 195

CHAPTER 10 ▪ The Smart Phones That Can Handle Fluids 199

CHAPTER 11 ▪ Fluid FX: Version 2.0 of Autodesk Fluid 203

CHAPTER 12 ▪ Show Time! MAYA Fluid Effects 205

CHAPTER 13 ▪ Fluids on Arbitrary Surfaces 213

CHAPTER 14 ▪ Control Freaks! How to Make Fluids Do What We Want 219

14.1 SHOOTING CANNONBALLS IN TWO DIMENSIONS 222

14.2 COMPUTER OPTIMIZERS 223

14.3 THE AD, THE JOINT, AND THE PATH BACK TO THE OPTIMIZER 233

CHAPTER 15 ▪ Real Experiments, Computer Experiments, and Validation 239

15.1 SPHERES ARE SUCH A DRAG 240

15.2 CURLY FLOWS BEHIND SPHERES, WAVY FLOWS IN TUBES, AND TURBULENT PLUMES BETWEEN PLATES 242

15.2.1 Experiment Number One: Curly Flows behind Spheres 243

15.2.2 Experiment Number Two: Wavy Flows in Tubes 243

15.2.3 Experiment Number Three: Turbulent Plumes between Plates 245

CHAPTER 16 ▪ Epilogue: Let’s Call It Quits 249

Preface

> What I cannot create, I do not understand.
>
> RICHARD FEYNMAN (FAMOUS AMERICAN PHYSICIST AND NOBEL LAUREATE)

> Lectures which really teach will never be popular; lectures which are popular will never teach.
>
> MICHAEL FARADAY (FAMOUS ENGLISH EXPERIMENTALIST AND SCIENTIST)

> A mathematical theory is not to be considered complete until you have made it so clear that you can explain it to the first man whom you meet on the street.
>
> DAVID HILBERT (FAMOUS GERMAN MATHEMATICIAN)

I have no formal background in fluid dynamics. I am not an engineer nor do I have a specialized degree in the mathematics or physics of fluids. I am fortunate that I did not have to carry that baggage around. On the other hand, I *do* have degrees in pure mathematics and computer science and have an artsy background. More importantly, I have written computer code that animates fluids.*

I wrote code. That is the bottom line.

* Other stuff as well of course. Chess games, Pac-Man rip offs, writing code to make money during summer jobs, ray tracers, particle system simulations, surface modeling, and more recently a unified dynamics solver called *Nucleus*. That is the beauty of writing code. The computer can be taught to do all sorts of things and you can do it anywhere. All you need is a laptop with an Internet connection. Sort of like mathematics. A mathematician is basically a clever mechanism that turns coffee into theorems. Coding and mathematical work can be done anywhere: the Copacabana Beach of Rio Janeiro (Brazil) or a remote cabin in Northern Canada.

I did not just download some code from the Internet and mash it together. Therefore, I think I understand what I am talking about. Of course, it is based on previous work. I did not invent the theories and concepts behind it. Research is a process.

You cannot fool a computer.

You can fool students, colleagues, or friends but not a computer. Why? A computer is like the best pet ever, it is wickedly fast and always obeys. Creating computer code is the ultimate test whether you understand something or not. Teach it to the computer and you will understand it. If you work for a company, you also have to deal with marketing, public relations, sales, and the customer feedback cycle.

Customer feedback is brutally honest. They pay you money for your software and they expect it to work. If it doesn't work, they will let you know. It either works or it doesn't. Postmodern literature debates, on the other hand, are not like that. Everyone is right and everyone is wrong at the same time. But these debates fueled by coffee or wine might be more fun than spending hours fueled by coffee fixing computer code that has to work.

I like writing code. And I like fixing code as long as it is my code.

This book is written by a dilettante of some sorts. However, by writing computer code, I came to understand the dynamics of fluids. This book is not your usual fluid dynamics textbook full of clever equations. I know equations quite well and I love them. They helped me to understand fluids. But to most people, they look like strange hieroglyphs from some other universe.

That is how Chinese characters appear to me.

That is why fancy equations are not very helpful to most people. When I get an e-mail written entirely in Chinese characters, I just ignore it. What? The alternative is that I take a course to learn Chinese characters. That would be cool and help me the next time I travel to China. But that is going to be a lot of work and my time is limited. Besides, it is fun to be a stranger in a strange land.

I want this book to be accessible to people who are not experts in mathematics or physics. I also want to make this a fun book. That is how I do research. I like to have fun even at *work*. Some parts of this book have mathematics. My hope is that the math will be somewhat accessible to most people who are willing to *go along*. This is a common tactic in mathematics when reading a mathematical paper that is not in your area of expertise and trying to understand it. At first, you just try to get the gist of it. Then if it sounds cool, you go through the details. And if you really get excited, then you should try to write some code that implements the content of the paper.

I also provide one-paragraph summaries of the material after each section in this book. I found this to be a good practice when I was learning math and computer science at the university. What did I learn in a nutshell after a lecture? I picked up this practice from a math professor, Pierre de la Harpe, at the University of Geneva who started each lecture with a summary of the material he covered in the previous lectures.

This methodology really helps if you are taking tons of classes. I also tend to understand some material using different tools than what most people are used to. I like to repeat things from different points of view. I like to argue with myself. That way, I can be both right and wrong, just not at the same time. You can learn some interesting stuff from this process. It is also a good tactic before giving a talk. You will be ready to face most questions. No question is stupid, and some can point you in new directions. I am always open to exploring new ideas, and learning. You can never be the smartest or the most creative person in the room.

The main goal of this book is to show how to create computer code that animates the motion of fluids. Computer code will be included in this book. Readers can download the accompanying code and run it on their own computers.

My goal is actually more ambitious. I want programmers to use these codes as a starting point to create their own apps, games, and so on. In fact, some of my code has been available for over 10 years and many people have used it as a starting point to create their own games, fun demos, and apps. Even better would be if programmers rewrite the code in a completely different manner or in a completely different language. That would be so cool. I would like to challenge any reader to write a shorter version of my code in C that is still readable.

I did not want to write a *Fluid Dynamics for Dummies* style of book. This is because I think my readers are smart and creative people who want to know how fluids are simulated on a computer to create nifty animations. In fact, I want smart and creative programmers to read this book and extend the code to create novel applications. Basically, I want this book to inspire people to do their own stuff. To create and not take anything for granted.

Be a rebel.

Not to destroy, but to create.

I love it when I get an e-mail out of the blue from someone far from my home in Toronto, like India, pointing me to a web-based application they created that combines fluids and reaction-diffusion processes. And "wow" the program runs in a web page. And the person thanks me, too. How cool is that. My day is certainly made.

Since there are so many good technical books on fluid dynamics, I want this book to be different and less technical. I want this book to be a bridge between my two favorite fluid dynamics books:

An Album of Fluid Motion (Stanford University, California, 1982), assembled by Milton Van Dyke*

A Mathematical Introduction to Fluid Mechanics (Springer, 2000), written by Alexandre Chorin and Jerrold Marsden

The first book is a collection of photographs of famous fluid experiments. It is the perfect geek coffee table book.

The second book is a brilliant and concise mathematical introduction to fluid dynamics. This book clarified many obscure aspects of fluid dynamics for me through a rigorous mathematical treatment. Consider the following statement: "Pressure is the Lagrange multiplier derived from the divergence free constraint." If this statement makes sense to you, then Chorin and Marsden's book is for you. But please continue reading this book.

This book is also a personal account of how I have dealt with fluid dynamics. Obscure at first, leading to enlightenment followed by writing computer code.

I want to make the narrative of this book somewhat interesting. Research is not just a collection of impersonal facts that seem to come out of nowhere. The science of fluids was created or discovered, whatever, by real people. I think it is important to give homage to them.

---

* I have to tell this story. I ordered this book from Milton Van Dyke's website and did not have to pay for it when I ordered it. Only when I got the book through regular mail, did I see that there was a notice to send a check of some ridiculously low amount of money to a certain Milton Van Dyke living in Palo Alto, California. No one I know would rip off a person who put together such a beautiful book. Of course, I immediately sent the check to the address with "Thanks!" written on the back of the check.

Don't worry.

This book is not going to describe how many wives, husbands, lovers, or kids these scientists had. There are plenty of good books out there documenting their personal lives. I am not an expert in these matters anyway nor am I particularly interested in their extracurricular activities. It is all about their scientific achievements.*

This book is based on many talks I have given over the last 15 years. Which explains the somewhat informal colloquial style of the book. Also, I assume my readers have access to search engines as there is no exhaustive list of references. If you like something in the book and want to know more about it, research it. That is how I work these days. I do miss the days I had to go to libraries, however, that is just nostalgia.

The first time I gave a talk on this subject was at an annual computer graphics conference called SIGGRAPH† in Los Angeles in 1999. The paper I presented was called "Stable Fluids." That was a crazy conference for me. I had other talks on completely different subjects,‡ one in the same afternoon, and I had to lug my hardware, an SGI Octane, using a cab from my hotel room at the Westin Bonaventure up on Figueroa Avenue to the convention center at the bottom of Figueroa.

But it worked out.

This was one of the first times that fluid simulations were shown to react to user input in real time. There was applause. And then there was applause again two years later, again in Los Angeles, when I gave my demonstration of real-time fluids on a Pocket PC.§ I wrote this just for fun. The Pocket PC fits in your pocket (hence the name). Consequently, I was able to show my fluid creations everywhere: at parties, on the subway, or to my family living abroad in Europe.

Only later did people tell me that I disproved some skeptics, that my fast demonstrations two years earlier were only due to using fancy hardware.

* But still. Leonhard Euler, one of the heroes of this book, had 13 kids. Legend has it that he did his finest math while holding one of his babies on his lap.

† SIGGRAPH is an acronym for Special Interest Group on GRAPHics and Interactive Techniques. Since 1974, this conference has been held yearly in various places across North America. It is the Mecca for graphics guys. It is the most prestigious place to publish a paper in computer graphics. In 1999, there were over 40,000 attendees.

‡ "Subdivision Surfaces" and "Diffraction Shaders."

§ The Pocket PC was released by Microsoft in 2000 and powered by an operating system called Windows CE. I showed my demo on an iPAQ, which was created by Compaq at the time. In 2007, Apple came out with the iPhone/iPod. I will say more about that later in the book.

When people disagree and challenge your work, that is actually a good sign. If no one cares about your work one way or the other, then why bother doing it.

The skeptics thought my demonstrations were fast because I showed them running on an SGI Octane.* I only needed the Octane because it had three-dimensional hardware texture rendering capabilities. Oh, and also because I got it for free and it was the best workstation at the time. But man, was it ever heavy to carry around.† It wouldn't fit in your pocket. The only reason I wrote apps for a mobile device was because it was a cool thing to do. In the end, it showed that software sometimes overpowers brute-force hardware.

I did this fun stuff when I was living in Seattle, Washington, in the late 1990s. Just after the *grunge scene* left town and moved to Los Angeles.‡ At the same time, our animation software called MAYA was about to take off. We used to have an office in sunny Santa Barbara, California, where everyone is always happy. When I went there on a trip from rainy Seattle and showed them the real-time demos, they immediately wanted to put it into our MAYA software. This was in 2000. After a roller-coaster ride, the fluid solver finally made it into our MAYA 4.5 software in 2002 under the name of "Fluid Effects." This release would have been impossible without the help of many people from the MAYA development team in Toronto, Canada. They did most of the work, getting my research code into a real piece of software and adding their own secret sauces and spices.

Putting research code into a product takes time and a lot of effort. Trust me. But in the end, it made it into MAYA! We shipped it. And people are using it. We didn't waste our time on some vaporware to prove we were busy and hardworking. And more importantly, we had fun doing it. Well, at least I had fun doing it.

In 2008, I wrote a fluid app for the iPhone/iPod. I did this mainly because the iPhone supports OpenGL ES, has an accelerometer, and has a multitouch interface. All the things that I wanted to have in an

---

* At the time, I worked for SGI who actually made the Octane. At the time, it cost over $60,000. Now, you can buy one on eBay for about $200. Oh right: SGI stands for Silicon Graphics Incorporated. It used to be the hottest company in Silicon Valley, California. They are pretty much defunct right now. Google has taken over their old campus in Mountain View.

† An Octane weighs 24.3 kg (54 lb).

‡ There were still a lot of good bands like the "Murder City Devils" and "Rorschach Test," just to name some.

interactive fluid app in 2000. I will say more about this experience later in the book. In 2010, we released Fluid FX, an improved version of the original app.

To summarize: I planted a seed in the area of fluid animation. Others have too. But this is my story.

Jos Stam
Toronto, Ontario, Canada

Acknowledgments

WRITING A BOOK AND DOING RESEARCH IS NOT A LONELY ENDEAVOR. There are many people I want to thank. First, thanks to my wife of 24 years, Pamela, for all her support and love. She is my best friend. To my daughter Gillian because … because she is my daughter and she is sweet, smart, and beautiful. To my older brother Sim, who introduced me to science and computers. Without him, I would possibly have become a hobo artist in Amsterdam. My parents, Jos and At, and my two sisters, Nel and Go, of course gave me a wonderful kick-start in life. We have short Dutch names in our family. I do not even have a middle name.

I thank CRC Press, especially Rick Adams, for publishing this book. And of course my company Autodesk, which gave me the time to write the book. I especially thank Gordon Kurtenbach, Azam Khan, Francesco Iorio, and the rest of our research group. It is great to be part of such a cool team. Special thanks also to Cory Mogk for creating the cover using MAYA Fluid Effects. Thanks to the Fluid FX team here in Toronto for creating these cool apps, based on my solver after I created my own. I want to single out Dan Pressman and Sergey Buyanov. I also thank the MAYA team we have in Toronto and teams we used to have in Santa Barbara and Seattle. This was a huge team, and if I have to single out one individual, it would be Duncan Brinsmead.

I thank my high school and university friend Marcus Grote for all the stimulating intellectual discussions we used to have. But more related to the current work he pointed me to Chorin and Marsden's book. This book led me out of the confusion and darkness. Since his dad worked at CERN, we got a VIP tour of the future of computing in the early 1980s. Thanks to my final high school year math teacher Nicolas Giovaninni, who taught me the difference between fun math and boring math. All of a sudden, I was getting top grades in math! Thanks to Professor Eugene Fiume for helping me to get accepted at a top North American graduate school: the University of Toronto. Also thanks for always being available and supportive. Professor Fiume was also a good supervisor, letting me work on whatever I wanted and not minding that I showed up at noon and left at five in the afternoon.

Thanks to the cities of Toronto, Paris, Helsinki, and Seattle. That is where I did most of the research described in this book despite all the distractions these cities have to offer. I find it is a good thing to live in different countries. You get rid of any *national pride* and *provincialism* you might have.* That stuff gets in the way of free and creative thinking. Moving around also makes you humble as you have to learn a new language and adapt to a new culture. It also makes you intellectually richer in the end. Just be willing to learn.

I also want to thank the SIGGRAPH Computer Graphics Technical Achievement Award Committee for presenting me with an award in 2005, partly for the work described in this book. Thanks also to the Academy of Motion Picture Arts and Sciences for presenting a Technical Achievement Award for our MAYA implementation of fluids. I am of course leaving out many other people that have influenced me. You know who you are if you are reading this. Thanks to all of you.†

* Unless of course Oranje is playing.

† If you feel left out, check out the slide, yes one slide, of my award acceptance speech at the SIGGRAPH Conference in 2005. The link is http://www.autodeskresearch.com/pdf/talks/jos_award05.pdf. If you are still left out, I will put on my Canadian tuck hat and say, "Sorry, buddy, but thanks anyway."

Author

Jos Stam was born in the Netherlands and educated in Geneva, where he received dual bachelor degrees in computer science and pure mathematics. In 1989, he moved to Toronto, where he completed his master's and PhD in computer science. After that, he pursued postdoctoral studies as an ERCIM fellow at INRIA in France and at VTT in Finland. In 1997, Dr. Stam joined the Alias Seattle office as a researcher and stayed there until 2003, when he relocated to Alias's main office in Toronto. He now works as a senior research scientist as part of Autodesk's acquisition of Alias in 2006.

Dr. Stam's research spans several areas of computer graphics: natural phenomena, physics-based simulation, rendering, and surface modeling, especially subdivision surfaces. He has published papers in all of these areas in journals and at conferences, most notably at the annual SIGGRAPH conference. In 2005, Dr. Stam was presented with one of the most prestigious awards in computer graphics: the SIGGRAPH Computer Graphics Achievement Award. In addition, he won two Technical Achievement Awards from the Academy of Motion Picture Arts and Sciences: in 2005 for his work on subdivision surfaces and in 2007 for his work on fluid dynamics. Dr. Stam was also featured in a January 2008 *Wired* magazine article.

CHAPTER 1

Introduction

> Simple play is also the most beautiful. How often do you see a pass of forty meters when twenty meters is enough? Or a one-two in the penalty area when there are seven people around you and a simple wide pass around the seven would be a solution? The solution that seems the simplest is in fact the most difficult one.
>
> JOHAN CRUIJFF (DUTCH SOCCER LEGEND)

> Problems can be complicated. Solutions cannot.
>
> ADVERTISEMENT SPOTTED AT THE COPENHAGEN AIRPORT AFTER GIVING A TALK ON FLUID DYNAMICS AT DANSIS

> There are times when we're testing an actual explosion, and then there are times when we blow stuff up just because we can.
>
> JAMIE HYNEMAN (CO-HOST OF THE POPULAR SHOW *MYTHBUSTERS*)

Visually, fluids are everywhere and nowhere.

That might seem like a weird statement. But let me explain.

Fluids like water, honey, maple syrup, fire, explosions, and oil are everywhere. We can see, smell, and sometimes taste and touch them. On the other hand, air is nowhere to be seen. Air is also a fluid. We usually see the effect of motion of the air on other things like insects, dust, clouds, and the motion of trees. Air is this invisible thing that makes things move. Some fluids are like that. The astonishing fact is that all these fluidlike effects such as liquids and air can be described by a single formal framework called *fluid dynamics*.

We will get to that.

What is a fluid? It really is a technical word that defines a substance that can change its shape in a continuous manner. So, a rock is not really a fluid but when it is heated it becomes one, like lava spurting from a volcano.* It also releases heat that will affect the motion of the fluid that is air. Lava definitely can take many different shapes. In fact that is why fluids are cool. A lava lamp is a good example. The legendary Toronto rock band "Sucker Punch" even has a song called "Cool Like a Lava Lamp." No really, this song rocks.

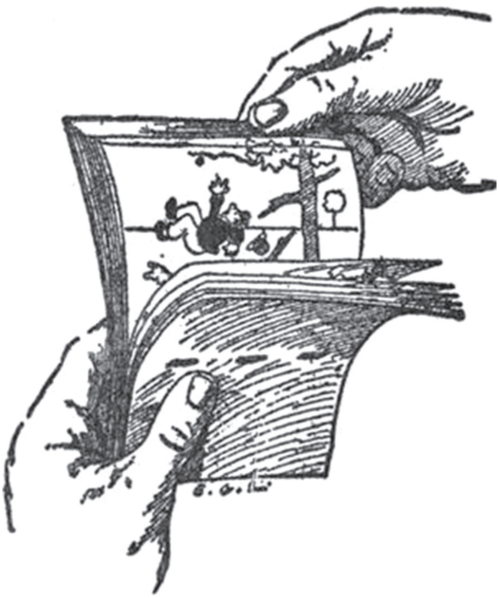
Since fluids are everywhere and nowhere, why do we want to animate them? There are many reasons actually. First, some fluid effects are hard to recreate, like nuclear explosions, tsunamis, and volcanic eruptions. In these cases, computer-generated fluids can create a multitude of catastrophic effects without anyone being hurt. Except, of course, for the animators who must put in long hours creating these effects, and often end up separating from their significant others. These animations can also create effects that have never been seen before or are impossible to create physically. In the special effects industry, they call actual physical shots with no computer graphics *practical*. I always found this to be an awkward word. Of course, some of these practical shots might be fixed "in post" with computer tools to get the perfect look. Practical means you actually blow stuff up, like they do in the popular television show *MythBusters*. Adam Savage and Jamie Hyneman, the two main hosts of this show, actually used to work for Industrial Light and Magic (ILM), which is a major special effects company based in San Francisco. Think *Star Wars*.

In addition to safety issues, there is another reason to use computers to create fluid animations: artistic control. In movies, people want fluids to behave in specific ways, which can be impossible or near impossible using *practical* methods. How would you get liquid horses coming out of a stream of water like in the *Lords of the Rings* movie using real footage? It might be possible but very improbable to achieve. When you simulate a fluid using computer animation you are playing God. You can vary control parameters

* On the other hand, rock formations can be considered in a sense as being very, very, very slow fluids. Think about the drift of continents that cause volcanoes and earthquakes.

to make the fluid do something you have in mind. Of course, this practice of animating fluids has been around since the early days of animation. Think of the classic Disney movies that featured fluids. However, this was a rather tedious method to create a convincing animation of a fluid. Back then every shot had to be hand drawn to create a series of what is called *key-frames*. To get an idea of the motion, artists would use a *flip book* basically flipping rapidly through the pages of a book sequentially. However, in this case, each page has a hand-drawn picture on it created by the artist. By a process of trial and error and intuition, a final fluidlike animation is created. Like that. Some artists were amazing at doing this.

One of the earliest flip books created is shown in Figure 1.1. It was invented in 1868 by John Barnes Linnett, an Englishman from Birmingham. He called it the *kineograph*, a fancy Latin word for *moving picture*. He filed a patent on it too. Why not? In those days there was a tremendous interest in animated pictures. It was not until December 28, 1895,* that the *Frères Lumière* from France showed their first motion



The kineograph

FIGURE 1.1 The first "flip book."

* Exactly 70 years to the date before I was born.

FIGURE 1.2 The Lumière brothers.

picture to the public. They are both portrayed in Figure 1.2 and one of the frames of their first movie is shown in Figure 1.3. Legend has it that the audience watching the movie for the first time was frightened. The sight of the train entering the station was so scary that most of the audience ran out of the cinema screaming. This short clip also had a fluid effect in it: the steam emanating from the locomotive. Interestingly, the brothers' last name *Lumière* means *light* in French, yes light, as in bright, not as in not being heavy.

The flip book and the cinema are good metaphors for computer animation. Indeed, the goal of computer animation is to create a sequence of pictures that give the illusion of movement. The bottom line is to fill an array of picture elements called *pixels* on a screen for every snapshot. Traditionally in computer graphics, things are animated in a virtual 3D world and then projected onto a screen. This is called the *rendering pipeline* and is depicted in Figures 1.4 and 1.5. The first figure is an engraving by the *German Leonardo* Albrecht Dürer. This is an early depiction of how perspective projection works and how it is used in art. Figure 1.5 shows the basic methodology of computer graphics. The ball,