MATH 255 Mathematics for Software Technology

2-2024

Course Syllabus 3(3-0-6)

Course Description:

Error of numerical methods, polynomial interpolation and curve fitting, root of equation of one variable and system of non-linear equations, system of linear equations and matrices, vector spaces and linear transformation.

Prerequisite: 206113 or 206161

Instructors:

Dr. Morrakot Khebchareon Office: MB2203 Email: mkhebcha@gmail.com

Class Meeting:

Tuesdays & Fridays 11.00-12.30

Course Assessment:

Midterm Exam	40%	Quizzes and Homework	20%
Final Exam	40%		

Letter Grade	A	B+	B	C+	C	D+	D	F
Score	80-100	75-79.99	70-74.99	65-69.99	60-64.99	55-59.99	50-54.99	0- 49.99

Important Notes:

1. Quizzes will be given during the class period.
2. There may be occasional homework assignment which requires submission but may not be graded.

Textbooks:

1. Cheney,W. and Kincaid, D.,,Numerical Mathematics and Computing
2. Johnson R.L.W., Dean Riess and Jimmy T.Arnold,,Introduction to Linear Algebra
3. Kharab, A. and Guenther, R.B.,,An Introduction to Numerical Methods: A MATLAB approach
4. Linz,P. and Wang,R.,,Numerical Methods: An Introduction to Scientific Computing using MATLAB
5. Penney,R.,,Linear Algebra: Ideas and Applications

Course Outline & Schedule:

Week	Day	Date	Topics
1	Tue	Nov 12, 2024	**Error of numerical methods** : Type of error, Truncation error and round off error, Numerical accurace, Dispersion of error, Floating point number
	Fri	Nov 15, 2024	
2	Tue	Nov 19, 2024	
	Fri	Nov 22, 2024	**Polynomial interpolation and curve fitting**: Type of interpolation and curve fitting, Lagrange polynomial, Divided difference, Curve fitting by least square method
3	Tue	Nov 26, 2024	
	Fri	Nov 29, 2024	
4	Tue	Dec 3, 2024	
	Fri	Dec 6, 2024	✓
5	Tue	Dec 10, 2024	
	Fri	Dec 13, 2024	
6	Tue	Dec 17, 2024	**Root of equation of one variable and system of non-linear equations**: Bisection method Secant method, Newton-Raphson's method, Iterative methods
	Fri	Dec 20, 2024	
7	Tue	Dec 24, 2024	
	Fri	Dec 27, 2024	
Mid-Term Exam: January 7, 2025 Time:15.30-18.30			
8	Tue	Jan 14, 2025	**System of linear equations and matrices**: Matrices and operations, Additive and product rule of matrices, System of linear equations, Gauss-Jordan elimination , Numerical method for solving system of linear equations - Direct methods - Iterative methods
	Fri	Jan 17, 2025	
9	Tue	Jan 21, 2025	
	Fri	Jan 24, 2025	
10	Tue	Jan 28, 2025	
	Fri	Jan 31, 2025	
11	Tue	Feb 4, 2025	
	Fri	Feb 7, 2025	
12	Tue	Feb 11, 2025	**Vector spaces and linear transformation**: Vector spaces, Properties of linear transformation, Algebra of linear transformation, Geometry of linear transformation
	Fri	Feb 14, 2025	
13	Tue	Feb 18, 2025	
	Fri	Feb 21, 2025	
14	Tue	Feb 25, 2025	
	Fri	Feb 28, 2025	
15	Tue	Mar 4, 2025	
	Fri	Mar 7, 2025	
Final Exam: Friday March 21, 2025 Time: 12.00 – 15.00			

Topics

Error of numerical methods : Type of error, Truncation error and round off error, Numerical accuracy, Dispersion of error, Floating point number

Base 10 Positional number

$0, 1, 2, 3, \ldots, 9$

$$23.25 = 2\times10^{1} + 3\times10^{0} + 2\times10^{-1} + 5\times10^{-2}$$

↑ decimal point

Base 2 : 0, 1

$$101.1001 = 1\times2^{2} + 0\times2^{1} + 1\times2^{0} + 1\times2^{-1} + 0\times2^{-2} + 0\times2^{-3} + 1\times2^{-4}$$

$$= 4 + 0 + 1 + 0.5 + 0 + 0 + \boxed{\frac{1}{2^4}} = 0.0625$$

$$= 5.5625$$

$2 \mid 5$

$2 \mid 2$ remainder 1

1 remainder 0

$(101)_2$

$(110101)_2 = 1\times2^5 + 1\times2^4 + 1\times2^2 + 1\times2^0$

$5\,4\,3\,2\,1\,0$

$= 32 + 16 + 4 + 1$

$= 53$

2 | 53		
2 | 26	1	
2 | 13	0	
2 | 6	1	
2 | 3	0	
1	1	

$53 = (110101)_2$

Fractional part of number

$0.5625 = (0.1001)_2$

0.5625×2

1.125×2

0.25×2

0.5×2

1

$0.5625 = (0.1001)_2$

$(0.1)_{10} = 0.000110011001100110011001100110011...$

$0.1 \times 2 = 0.2$
$0.2 \times 2 = 0.4$
$0.4 \times 2 = 0.8$
$0.8 \times 2 = 1.6$
1.6×2

$1.2 \times 2 = 0.4$
$0.4 \times 2 = 0.8$
$0.8 \times 2 = 1.6$
1.6×2

$1.2 \times 2 = 0.4$
$0.4 \times 2 = 0.8$
$0.8 \times 2 = 1.6$
1.6×2

$1.2 \times 2 = 0.4$
$0.4 \times 2 = 0.8$
$0.8 \times 2 = 1.6$
1.6×2

$1.2 \times 2 = 0.4$
$0.4 \times 2 = 0.8$
$0.8 \times 2 = 1.6$
1.6×2

$1.2 \times 2 = 0.4$
$0.4 \times 2 = 0.8$
$0.8 \times 2 = 1.6$
1.6×2

...

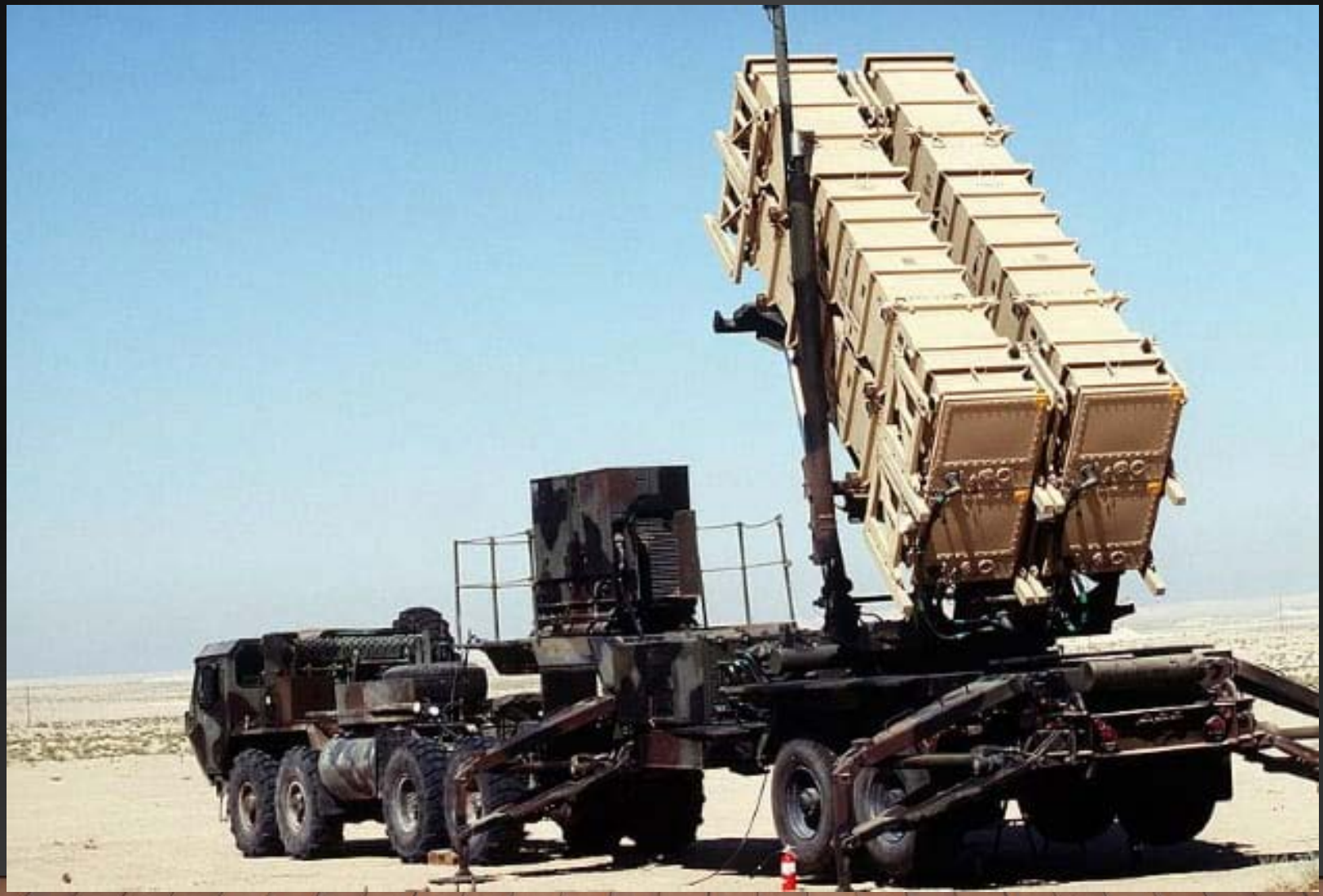
10 Simple But Costly Math Errors In History

by Oliver Taylor

A lot of students hate math, even though the world runs on mathematics. Huge sums of money and equipment worth millions of dollars have been lost simply because someone made some tiny math mistake. Computers also have made errors like this, too, but that can still be blamed on humans since we did the programming.

To be clear, some calamities were caused by errors involving complex math, which is arguably more forgivable. However, the ones we have here are mostly ridiculously simple mistakes that even an elementary school student might not have made.

Gulf War Scud Missile Attack



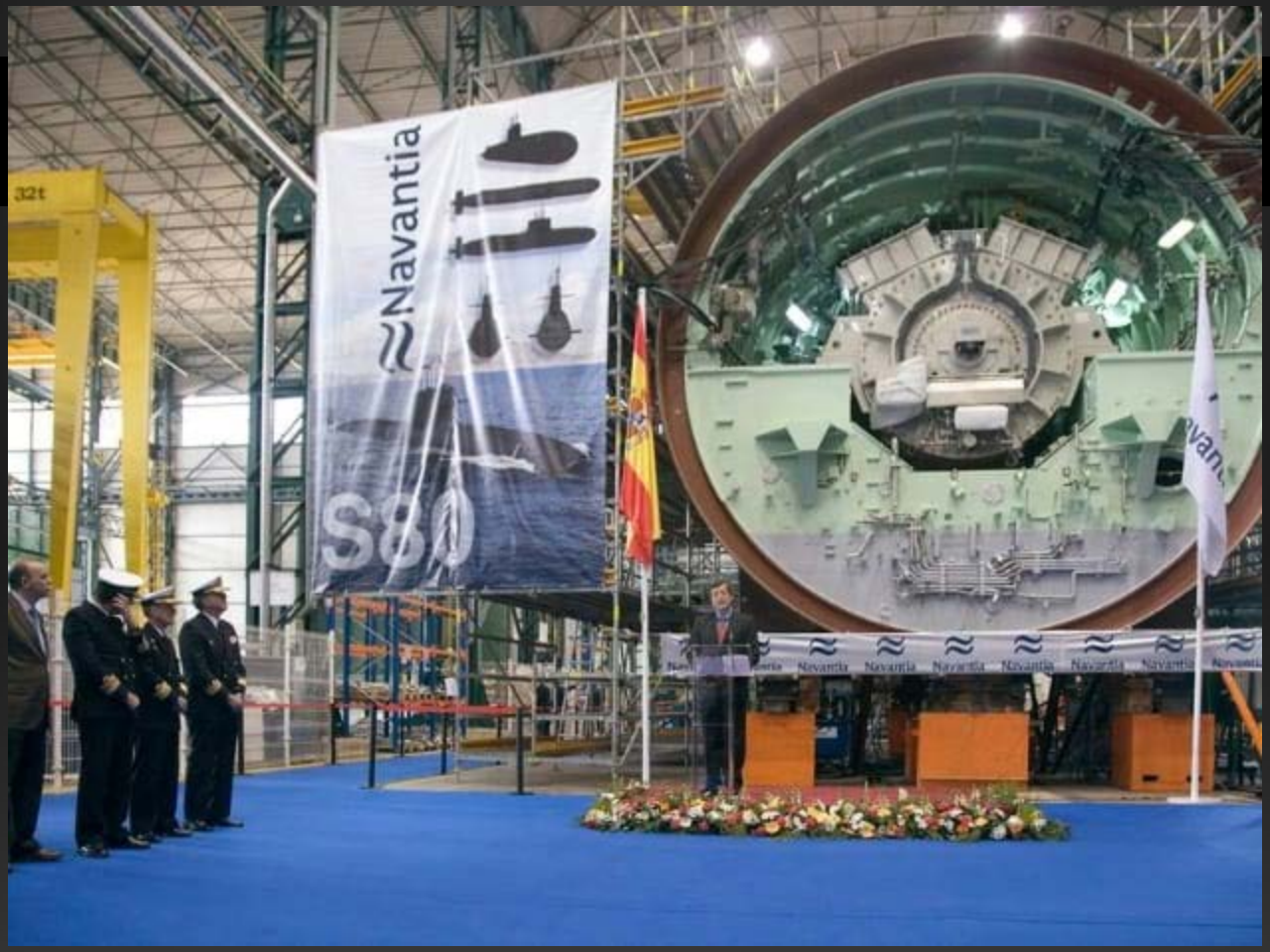
10 Simple But Costly Math Errors In History - Listverse

On February 25, 1991, an Iraqi "Scud" missile struck a US Army base in Dharan, Saudi Arabia, killing 28 soldiers and injuring 100 others. The disaster was surprising, considering that the base was protected by a Patriot missile air defense system. Investigations revealed that the system did not attempt to intercept the Scud.

The error was traced to the software powering the clock of the system. The clock recorded time in deciseconds (one tenth of a second) but stored that data as an integer. It converted the time into a 24-bit floating point number to do this. However, rounding the times in order to convert them led to gradually increasing inaccuracy as the system operated. As a result, the system was not able to intercept missiles after 20 hours of continuous use.

At the time of the attack, the Patriot missile battery in question had been operating for 100 hours. The time disparity was such that it looked in the wrong part of the sky for the incoming missile and therefore found no target. The US Army had been made aware of this software issue and had released an update on February 16. The update only reached the Dharan base on February 26, a day after the attack.[1]

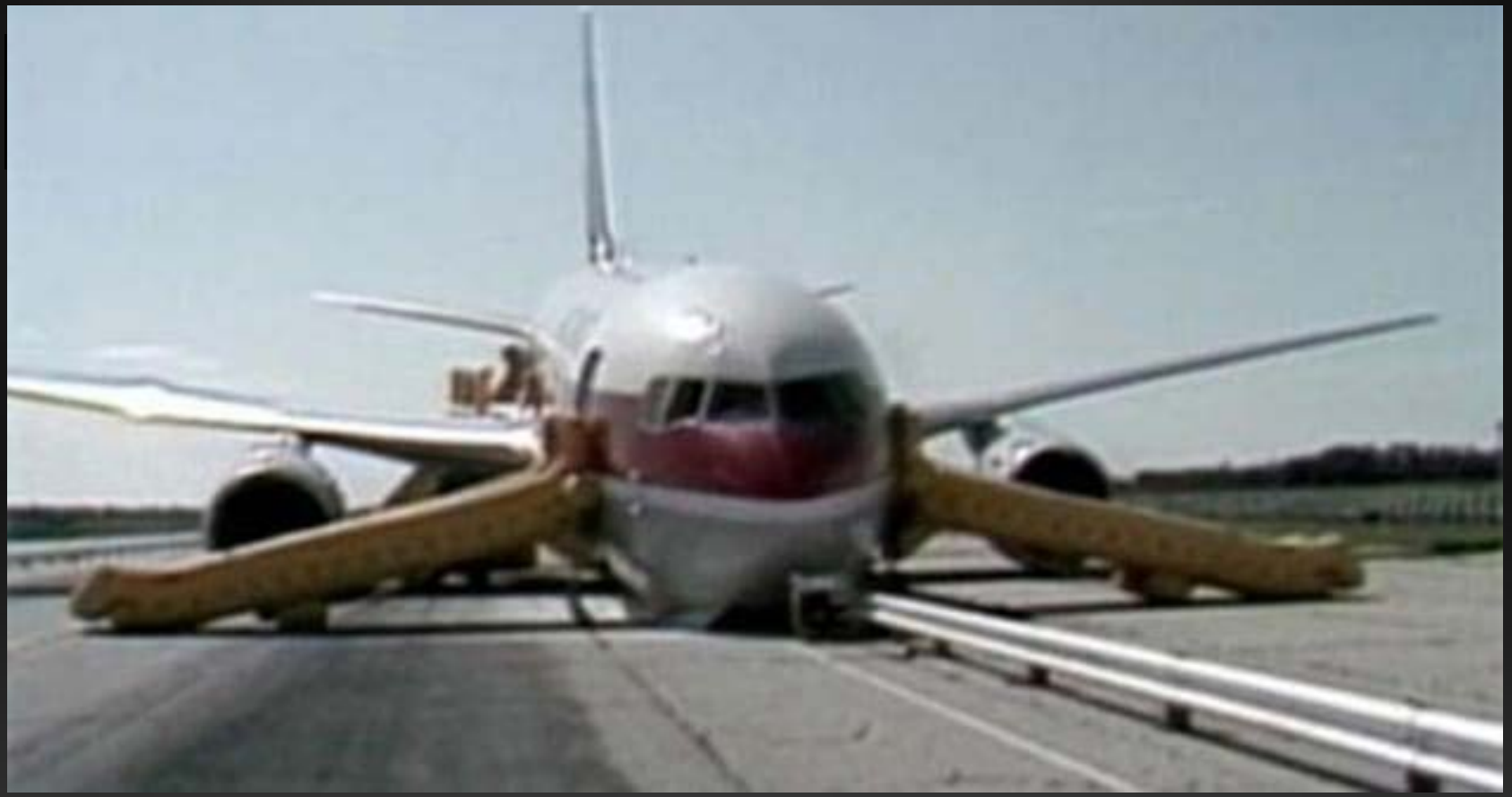
Spain's S-80 Submarine Program



In 2003, Spain launched the $2.7 billion S-80 submarine program to build four diesel-electric submarines for the Spanish navy. Spain had almost completed one of the submarines in 2013, when it discovered that the sub was 70 tons heavier than it should have been. The Spanish navy feared the submarine would never surface if it went underwater.

The submarine ended up heavy after someone put a decimal point in the wrong spot during calculations. No one discovered the error until the first submarine was completed, and the other three were already under construction. Spain later signed a $14 million deal with Electric Boat of Groton, Connecticut, to help them reduce the weight of the 2,200-ton submarine.[2]

Air Canada Flight 143



In July 1983, an Air Canada Boeing 767 flying from Ottawa to Edmonton with 69 passengers and crew had to crash-land after running out of fuel at 12,500 meters (41,000 ft). The engines suddenly lost power, and the airplane started gliding to the ground. It glided for 100 kilometers (60 mi) before landing in Gimli, Manitoba.

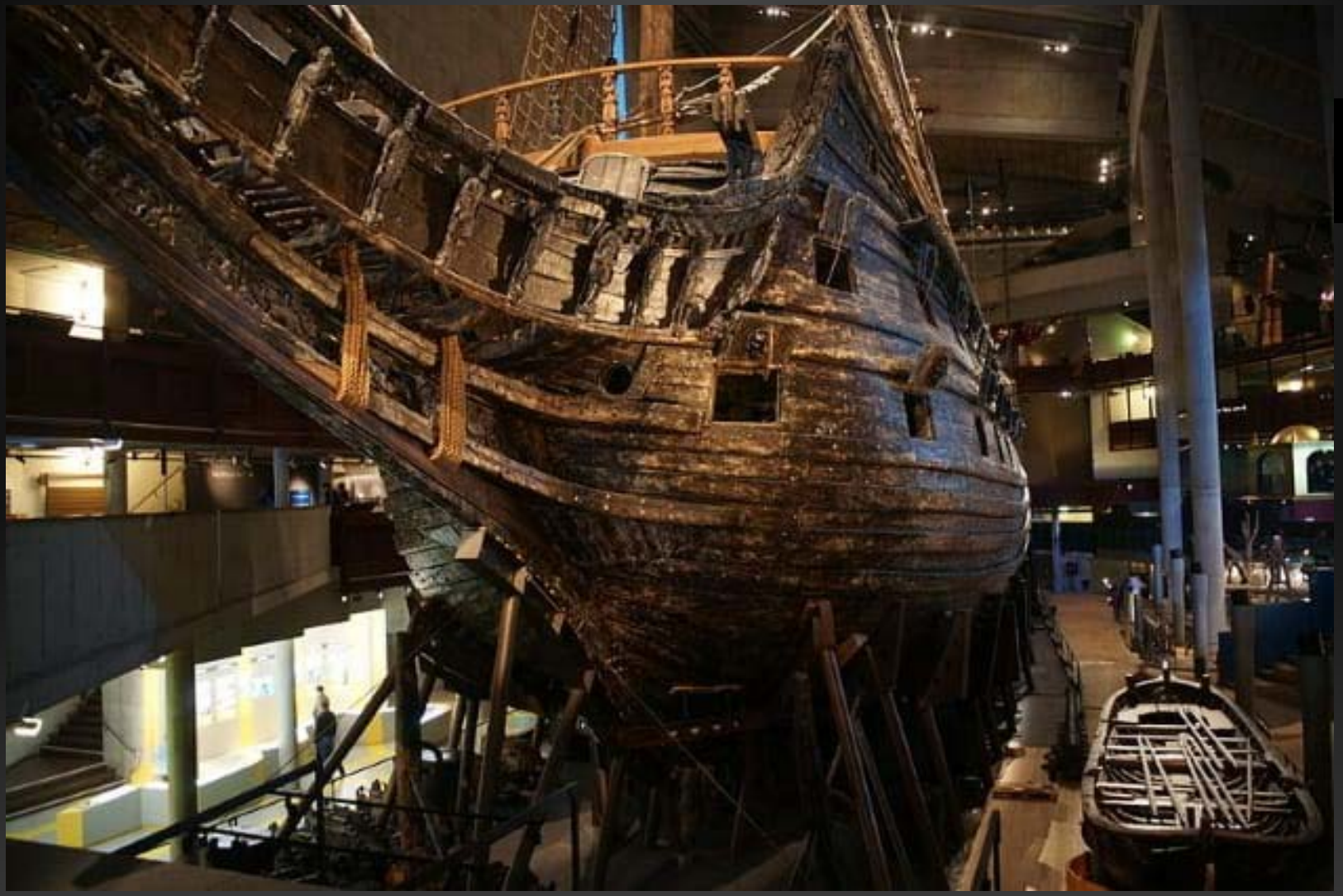
It came down on a racetrack that had originally been a runway. Luckily, there were no deaths. However, two people had minor injuries, and the nose gear was destroyed. This landing earned Flight 143 the nickname of "Gimli Glider."

The accident was traced to a conversion error. Air Canada used the imperial system of measurement but was converting to the metric system, which this Boeing 767 already used. Air Canada ground crews had used the imperial system when they refueled the airplane. They measured the fuel in pounds instead of kilograms.

One kilogram equals 2.2 pounds. This meant the airplane had only around half the amount of fuel it required to complete the flight. The pilots did not notice the discrepancy because the fuel gauge was not working. Ground crews used drip sticks to measure the fuel at the time they filled the tanks.

Interestingly, the ground crews made the mistake twice. The first was in Montreal, and the second was in Ottawa. The airplane made the Montreal-to-Ottawa flight without incident but literally flew into problems when it was flying from Ottawa to Edmonton.[3]

Sinking Of The *Vasa*

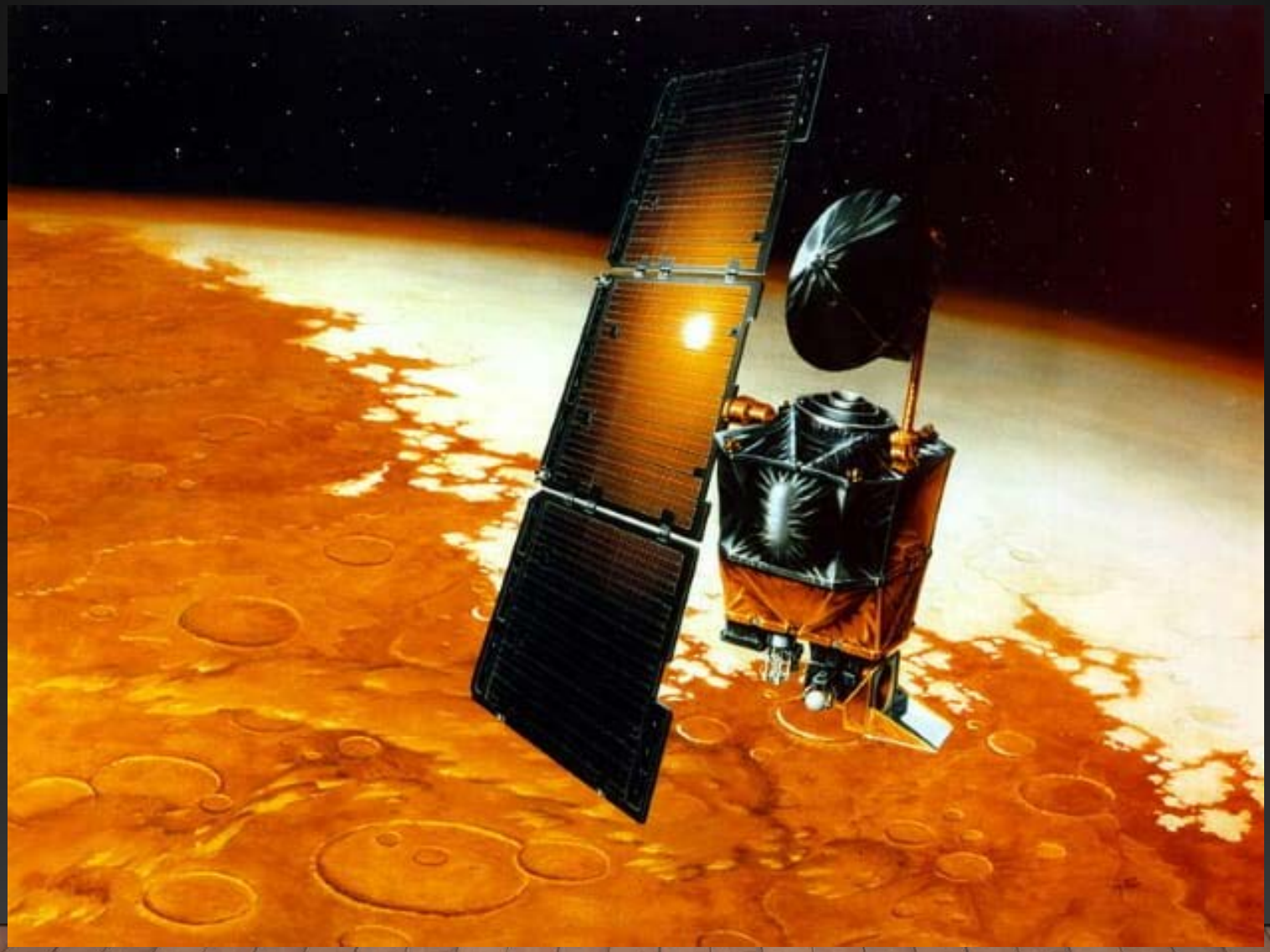


On August 10, 1628, Sweden launched a new, heavily armed, and large warship: the *Vasa*. The vessel had barely sailed for 20 minutes when it sank less than a mile from shore. Thirty people died in the sinking. The ship was later retrieved in the 20th century and is now held at the Vasa Museum.

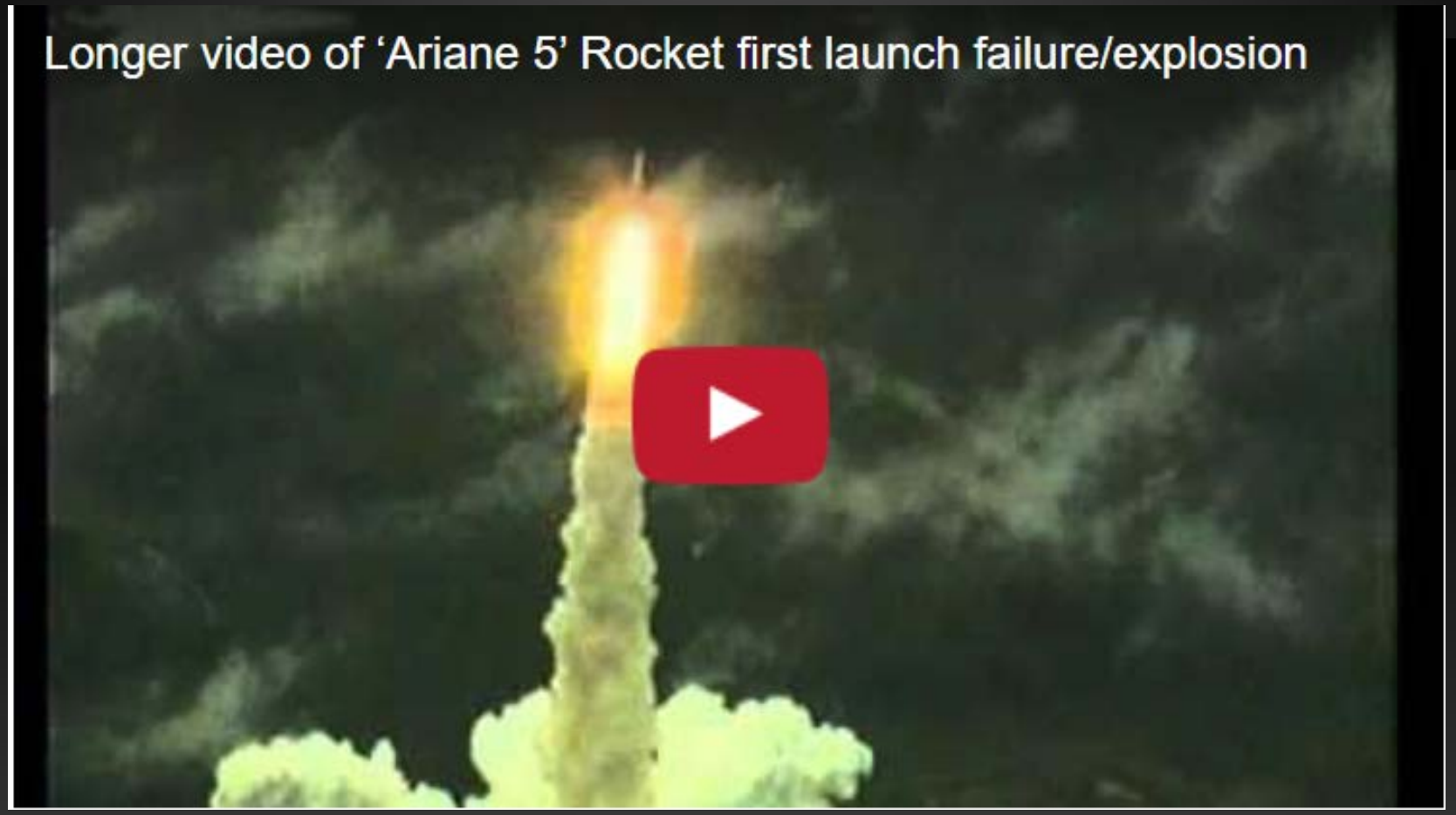
Historians measured the entire ship and discovered that its builders used two different units of measurement. One was the Swedish foot, and the other was the Amsterdam foot. A Swedish foot is 12 inches, while an Amsterdam foot is 11 inches.

The difference between both units of measurements caused one side to end up heavier than the other. This was why the ship leaned to one side and promptly sank after it was hit by two gusts of wind. Historians add that the effect of the wind was worsened by the fact that the top of the ship was heavier than its bottom.[4]

Mars Climate Orbiter Crash



Ariane 5 Rocket Explosion

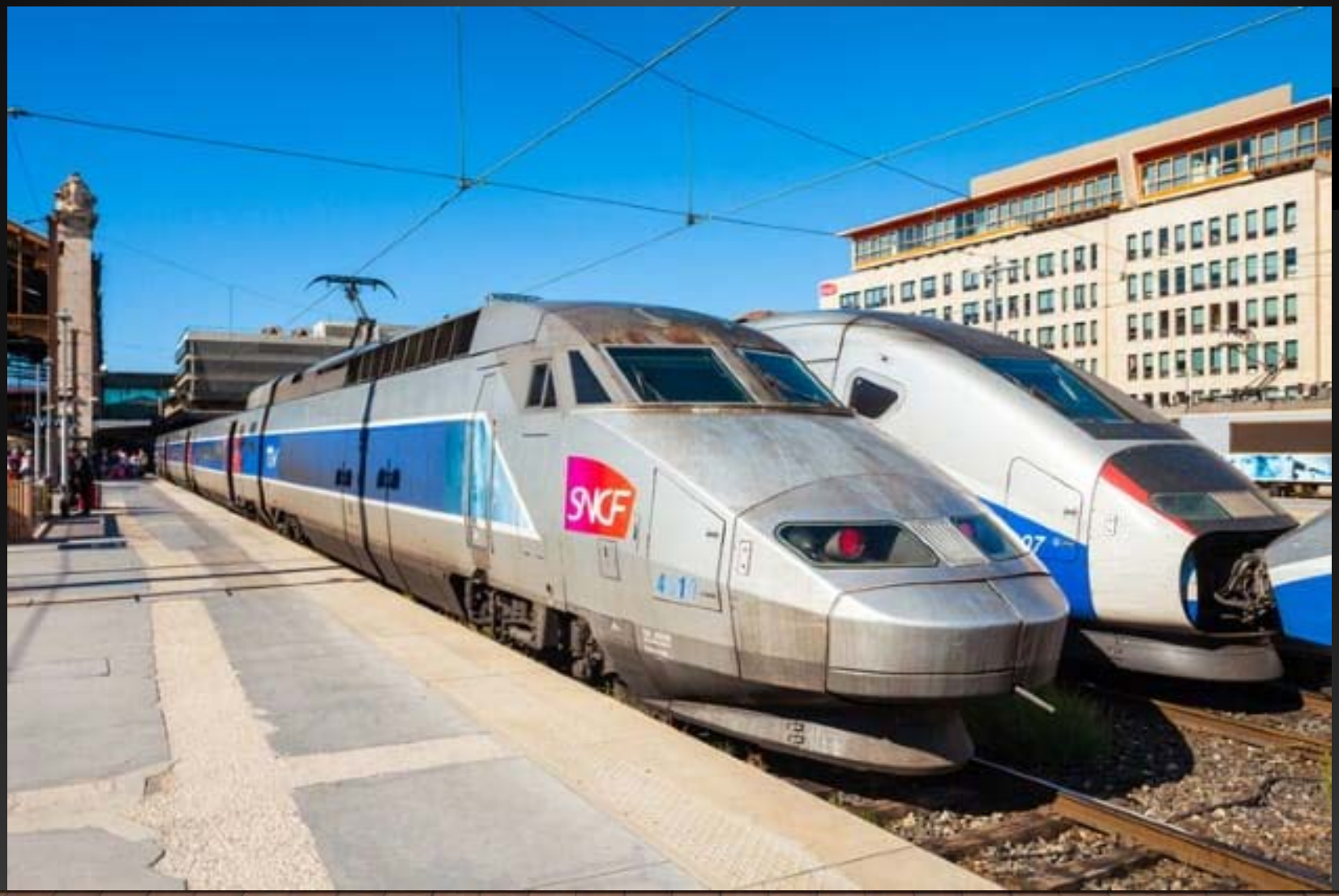


On June 4, 1996, the European Space Agency's Ariane 5 rocket exploded 37 seconds after takeoff. Onboard the spacecraft were four satellites. The rocket and satellites cost $370 million. The accident was traced to an integer overflow error in the software used for launching the rocket.

An integer overflow is a mathematical error that occurs when the figures generated by a system exceeds the memory of that system. The Ariane 5 operated on 16-bit software capable of storing figures up to 32,767. The rocket managed to generate figures way above that.

The European Space Agency used the same software they'd previously used in Ariane 4 rockets. They had problems with the Ariane 5 because it was faster than the Ariane 4. Faster means larger figures. The software could not handle the large readings, causing the rocket to go rogue. Ground control ordered it to self-destruct.[6]

France's Oversized Train Problem



In 2014, Societe Nationale des Chemins de Fer francai (SNCF), France's state railway operator, discovered its new high-speed trains were too wide for 1,300 stations across the country. The problem was that it had ordered 1,860 of the trains from Alstom of France and Bombardier of Canada. SNCF determined that it needed to reduce the width of the trains so that the stations could accommodate them. The error cost millions of euros.

The incident generated some displeasure in France; the transport minister referred to it as "comically tragic." *Canard Enchaine*, a weekly satirical paper, made a cartoon in which commuters on a platform were told to "pull in their stomachs" as one of the new trains approached the station.

The mistake happened because French train stations vary in size. SNCF knew this and had requested the Reseau ferre de France (RFF), which was in charge of the tracks, to measure the space around the tracks. SNCF and RFF ended up with some problems after it was realized that RFF had skipped 1,300 older stations in its initial calculations. These stations were narrower than others. It was too late, as some trains had been delivered, and more were under construction.[9]