Math.375 Fall 2005
I - Numbers and Formats

Vageli Coutsias

Introduction

– 1 + 1 = 0 or “machine epsilon”?

– » eps = 2.220446049250313e-016

How does matlab produce its numbers?

- Where we learn about number formats, truncation errors and roundoff

Matlab real number formats

» format long %(default for π)

pi = 3.14159265358979

» format short

pi = 3.1416

» format short e

pi = 3.1416e+000

» format long e

pi = 3.141592653589793e+000

Floating-point numbers

$$x = \pm\left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \frac{d_3}{\beta^3} + \cdots + \frac{d_t}{\beta^t}\right)\beta^e$$

β Base or radix
t Precision
[L,U] Exponent range

$$0 \le d_i \le \beta - 1, i = 1, \ldots, p; d_1 \ne 0$$

$$L \le e \le U$$

SYSTEM	Base	Precision	L(ow Exp)	U(pperExp)
IEEE SP	2	24	-126	127
IEEE DP	2	53	-1022	1023
Cray	2	48	-16383	16384
HP Calc	10	12	-499	499
IBM mainfr	16	6	-64	63

$$x = (-1)^s \cdot (0.d_1 d_2 \cdots d_t) \cdot \beta^e = (-1)^s \cdot m \cdot \beta^{e-t}$$

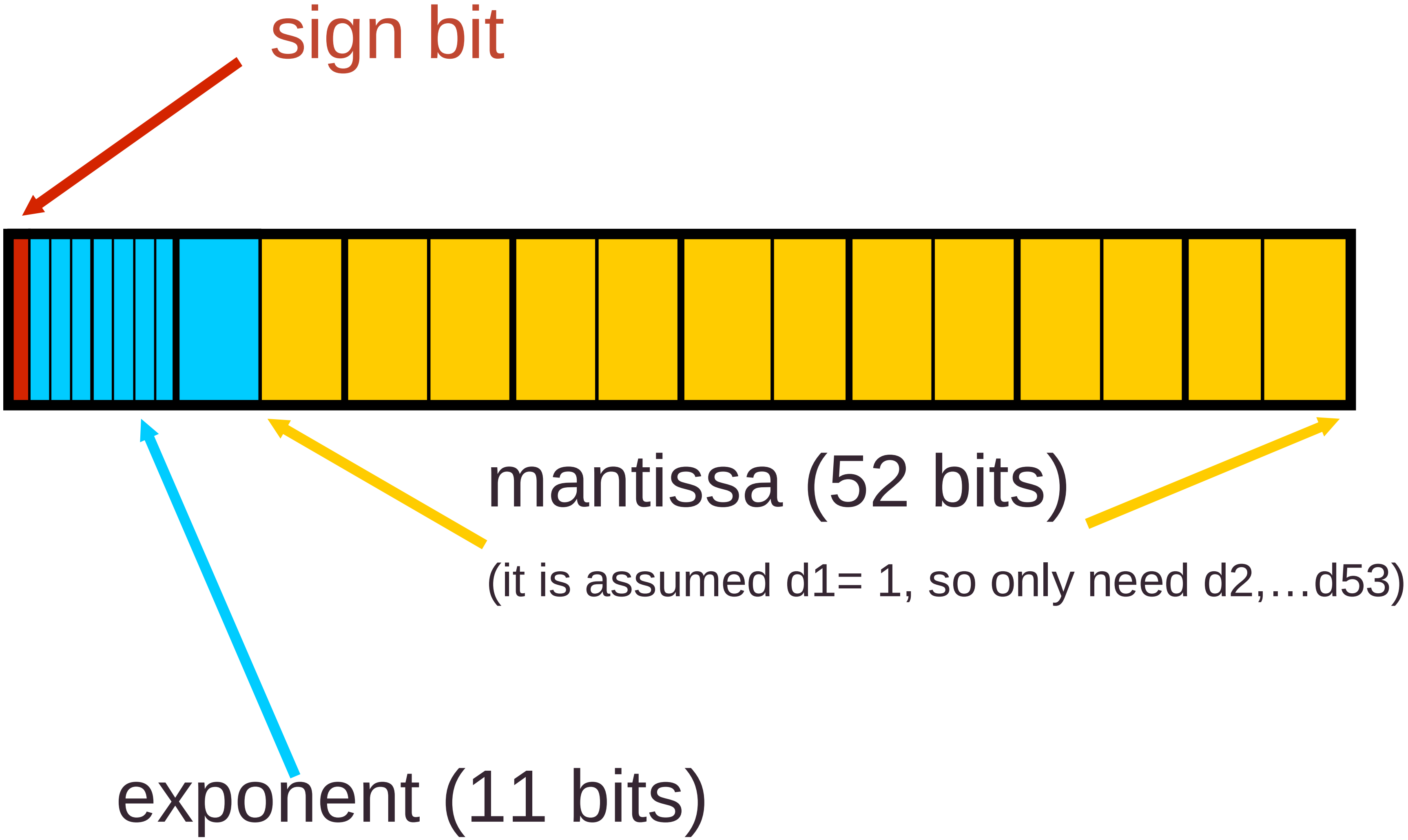
$$m = d_1 d_2 \cdots d_t \qquad d_1 \ne 0$$

The set **F** of f.p. numbers

- Basis β
- Significant digits t
- Range (U, L)

- $F(\beta, t, U, L)$: $F(2, 53, -1021, 1024)$

  is the IEEE standard


sign bit
mantissa (52 bits)
(it is assumed d1= 1, so only need d2,…d53)
exponent (11 bits)

IEEE double precision standard

If E=2047 and F is nonzero, then **V=NaN** ("Not a number")

If E=2047 and F=0 and S=0,(1) then **V=Inf**, **(-Inf)**

If E=0 and F=0 and S=0,(1), then **V=0,(-0)**

If 0<E<2047 then

V=(-1)**S * 2 ** (E-1023) * (1.F)

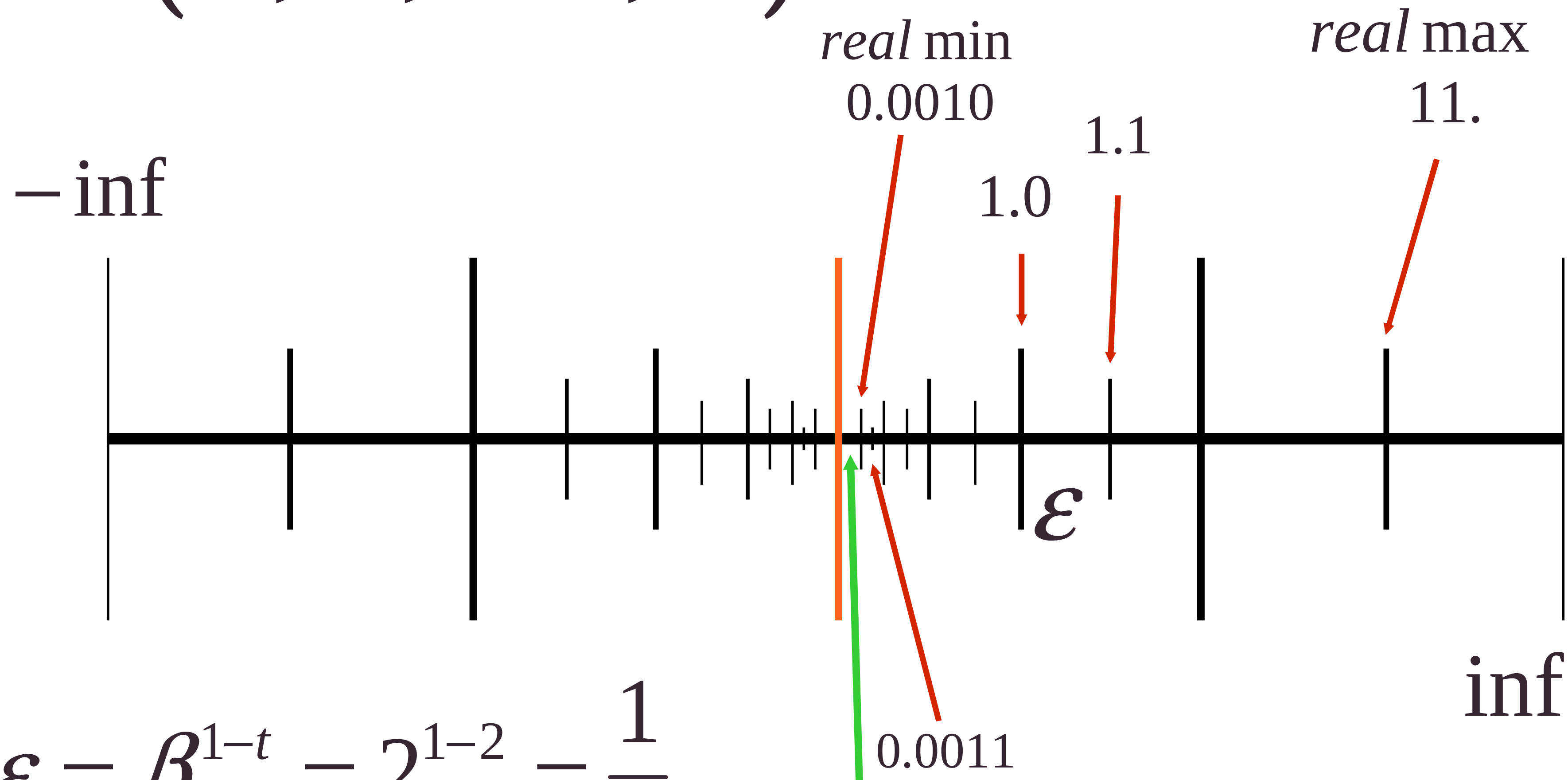
where "1.F" denotes the binary number created by prefixing F with an implicit leading 1 and a binary point.

If E=0 and F is nonzero, then

V=(-1)**S * 2 ** (-1022) * (0.F)

These are "unnormalized" values.

$$F(2,2,-2,2)$$



$$\varepsilon = \beta^{1-t} = 2^{1-2} = \frac{1}{2}$$

gap contains “unnormalized” values, as allowed, e.g., in IEEE standard

Floating point numbers

$$Underflow\_level := UFL = \beta^{L-1}$$

$$Overflow\_level := OFL = \beta^{U}(1-\beta^{-t})$$

$$\varepsilon := eps = \beta^{1-t}$$

The machine precision is the smallest nuber ε such that:

$$fl(1+\varepsilon) > 1$$

$$IEEE\_sp\_\varepsilon = 2^{-23} \approx 10^{-7} \qquad IEEE\_dp\_\varepsilon = 2^{-52} \approx 10^{-16}$$

Machine epsilon

- The distance from 1 to the next larger float

$$\varepsilon := eps = \beta^{1-t}$$

- Gives the relative error in representing a real number in the system F:

$$\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2}\varepsilon$$

(RELATIVE) ROUNDOFF ERROR

Machine epsilon computed

```
a = 1; b = 1;
while a+b ~= a
    b = b/2;
end
    b
% b = 1.110223024625157e-016
% shows that a+b = a is satisfied by
% numbers b not equal to 0
% here b = eps/2 is the largest such
% number for a = 1
```

- Overflow does not only cause programs to crash!

Arianne V’s short maiden flight on 7/4/96 was due to a floating exception.

FLOAT → INTEGER

- During the conversion of a 64-bit floating-point number to a 16-bit signed integer
- Caused by the float being outside the range representable by such integers
- The programming philosophy employed did not guard against software errors-a fatal assumption!

COMPLEX NUMBERS

- z = x + i*y
- x = Re(z)   is the real part
- y = Im(z)   is the imaginary part
- i^2 = -1   is the imaginary unit

- polar form $$z = \rho e^{i\vartheta} = \rho(\cos\theta + i\sin\theta)$$

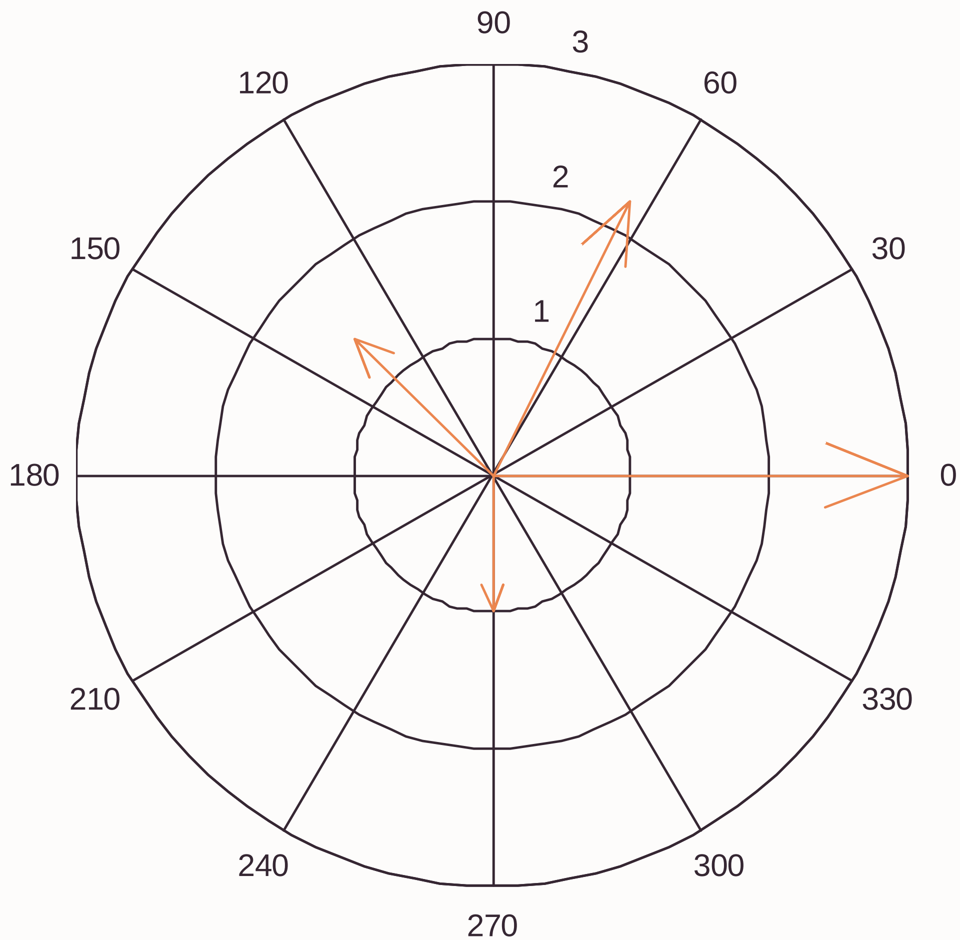
- complex conjugate $$\bar{z} = x - iy$$

- Matlab commands:

```
>>    z = 3+i*4
>>% Cartesian form:
        x = real(z);  y = imag(z)
>>% Polar form:
        theta = angle(z); rho = abs(z)
So:    z = abs(z)*(cos(angle(z)) + i*sin(angle(z)))

        x-i*y = conj(z)
```

The complex plane



z=[1+2*i, 3,-1+i, -i];

compass(z,'r')

defines an array of complex numbers which are plotted as vectors in the x-y plane

Roots of complex numbers

```
» x = -1 ;  x^(1/3)

ans =

           0.5000 + 0.8660i
```

Matlab assumes complex arithmetic, and returns automatically the root with the smallest phase angle
(the other two roots are

-1

and

0.5000 - 0.8660i

Types of errors in numerical computation

- Roundoff errors

  Pi = 3.14159

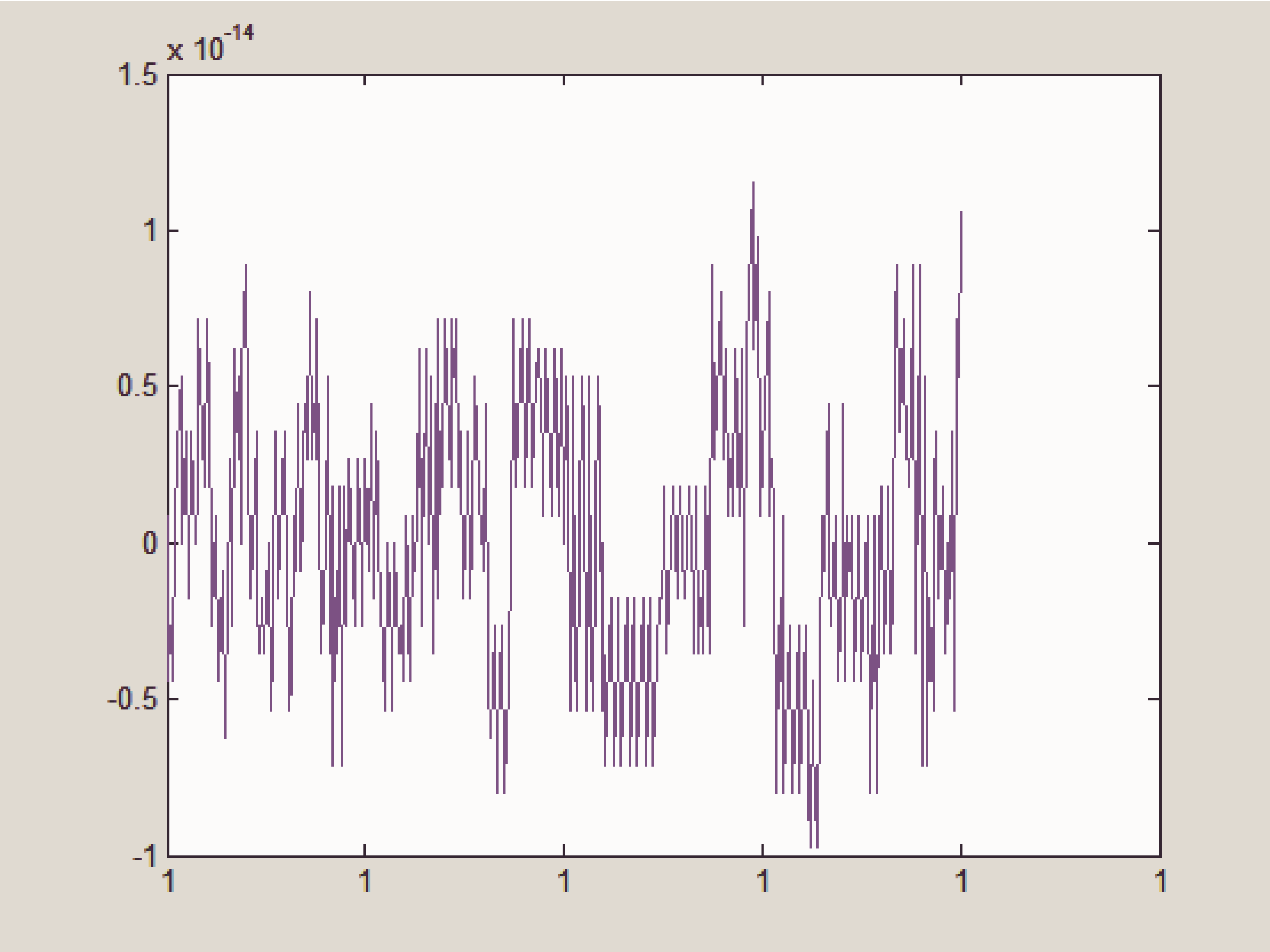
  Pi = 3.14159265358979323846 26
- Truncation errors

  Cos x = 1 – x^2/2

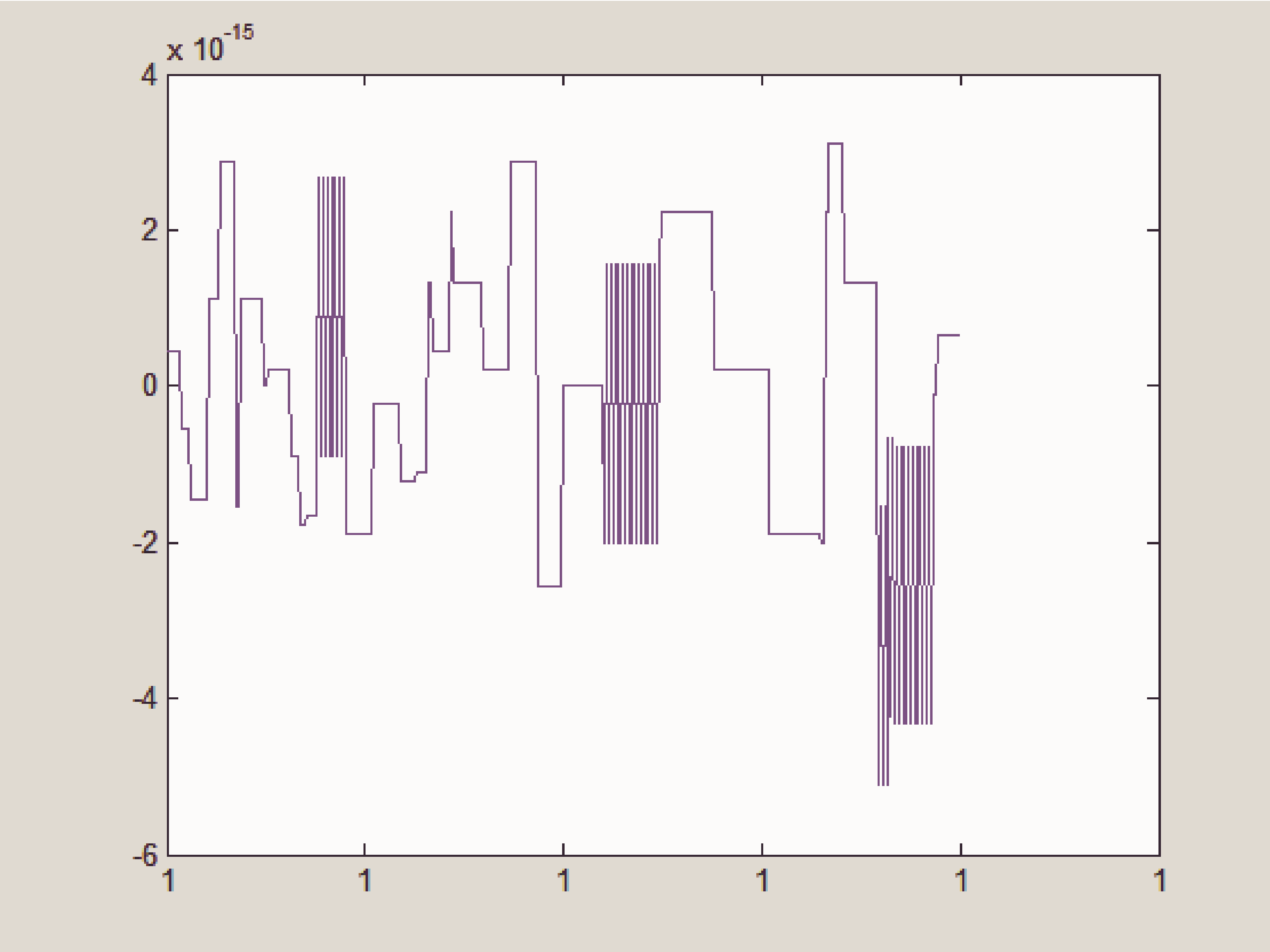
  Cos x = 1 – x^2/2 + x^4/4!

Errors usually accumulate randomly (random walk)

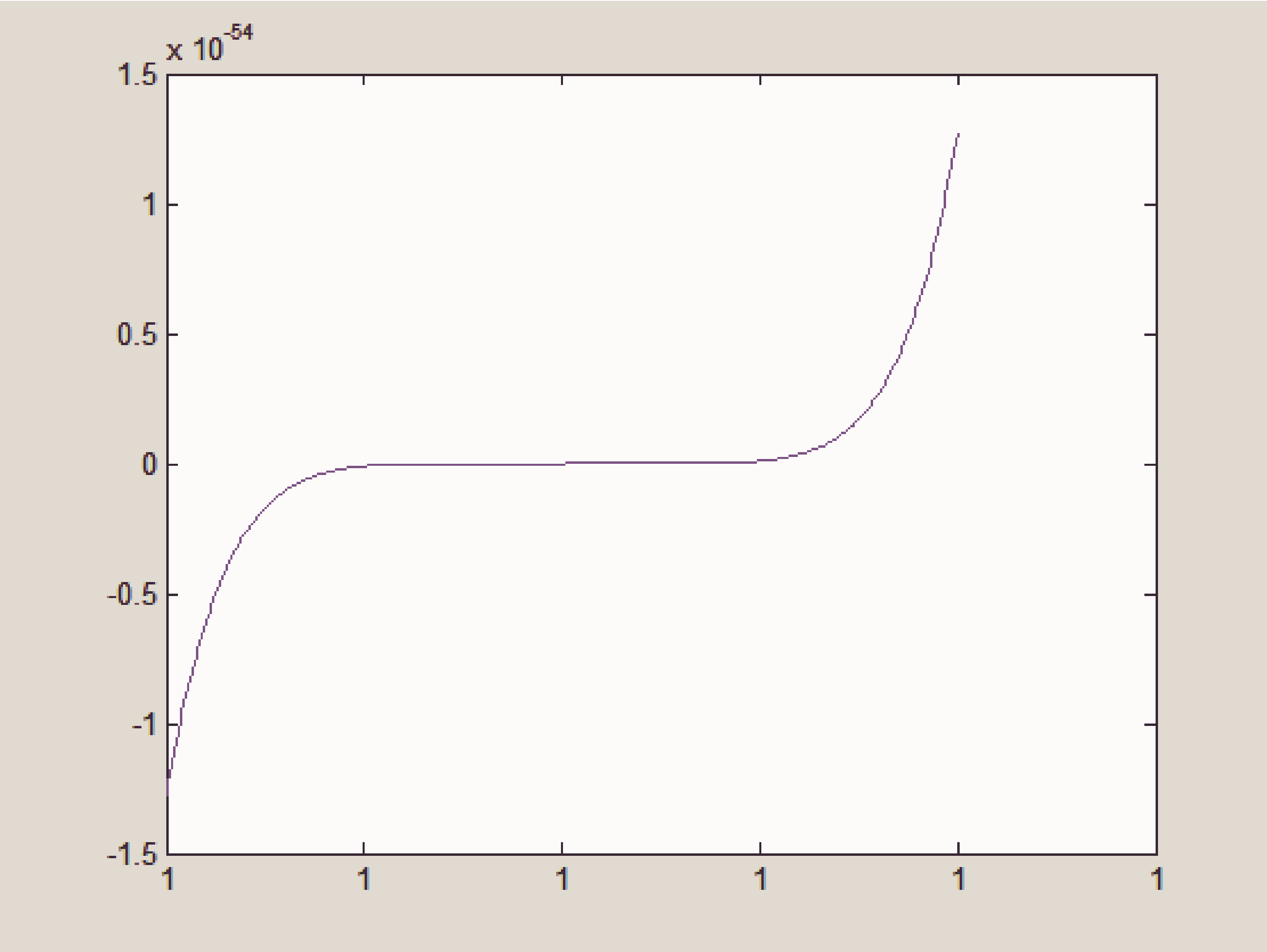
But they can also be systematic, and the reasons may be subtle!



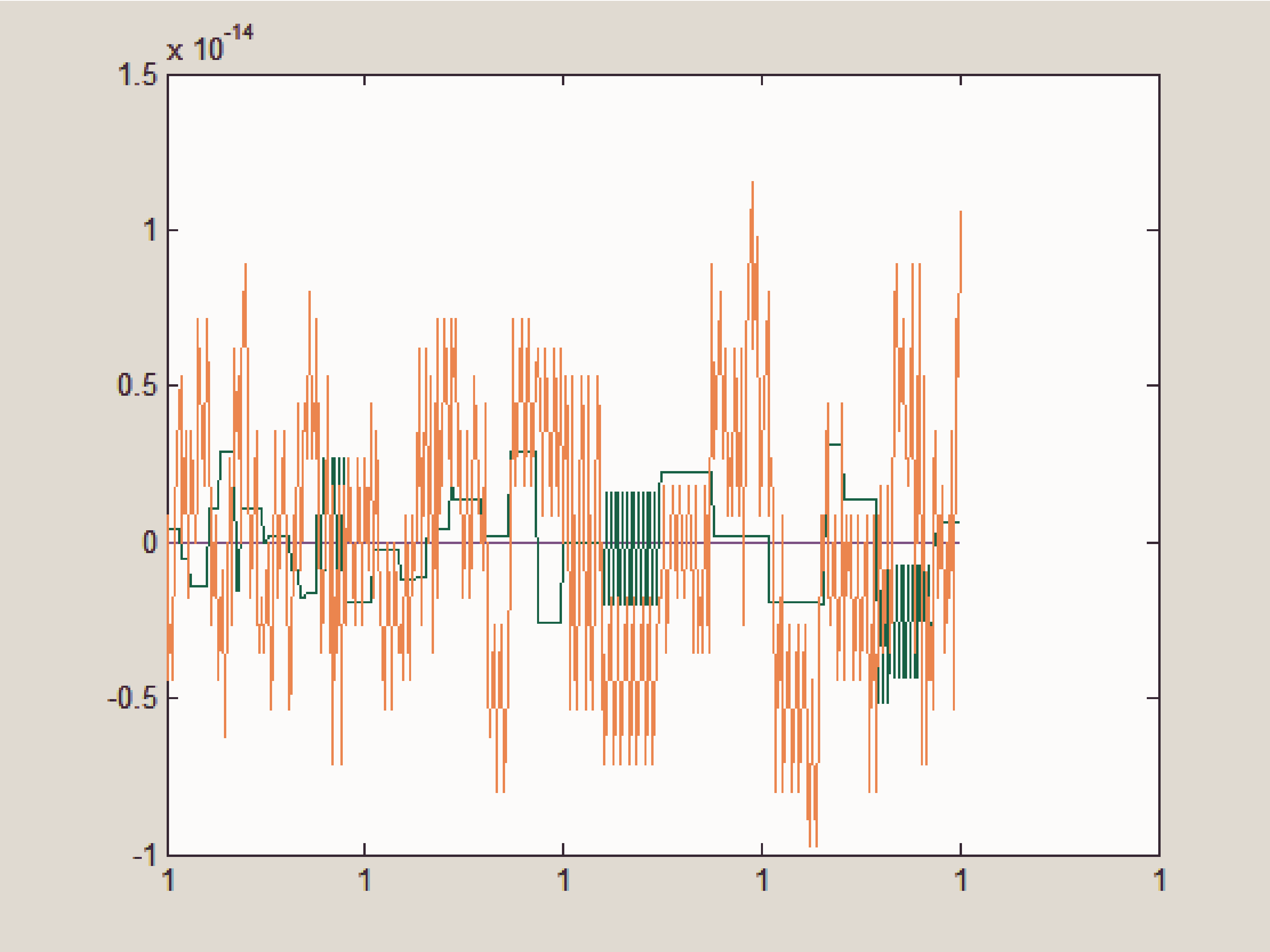

```
x = linspace(1-2*10^-8,1+2*10^-8,401);
f = x.^7-7*x.^6+21*x.^5-35*x.^4+35*x.^3-21*x.^2+7*x-1;
plot(x,f)
```



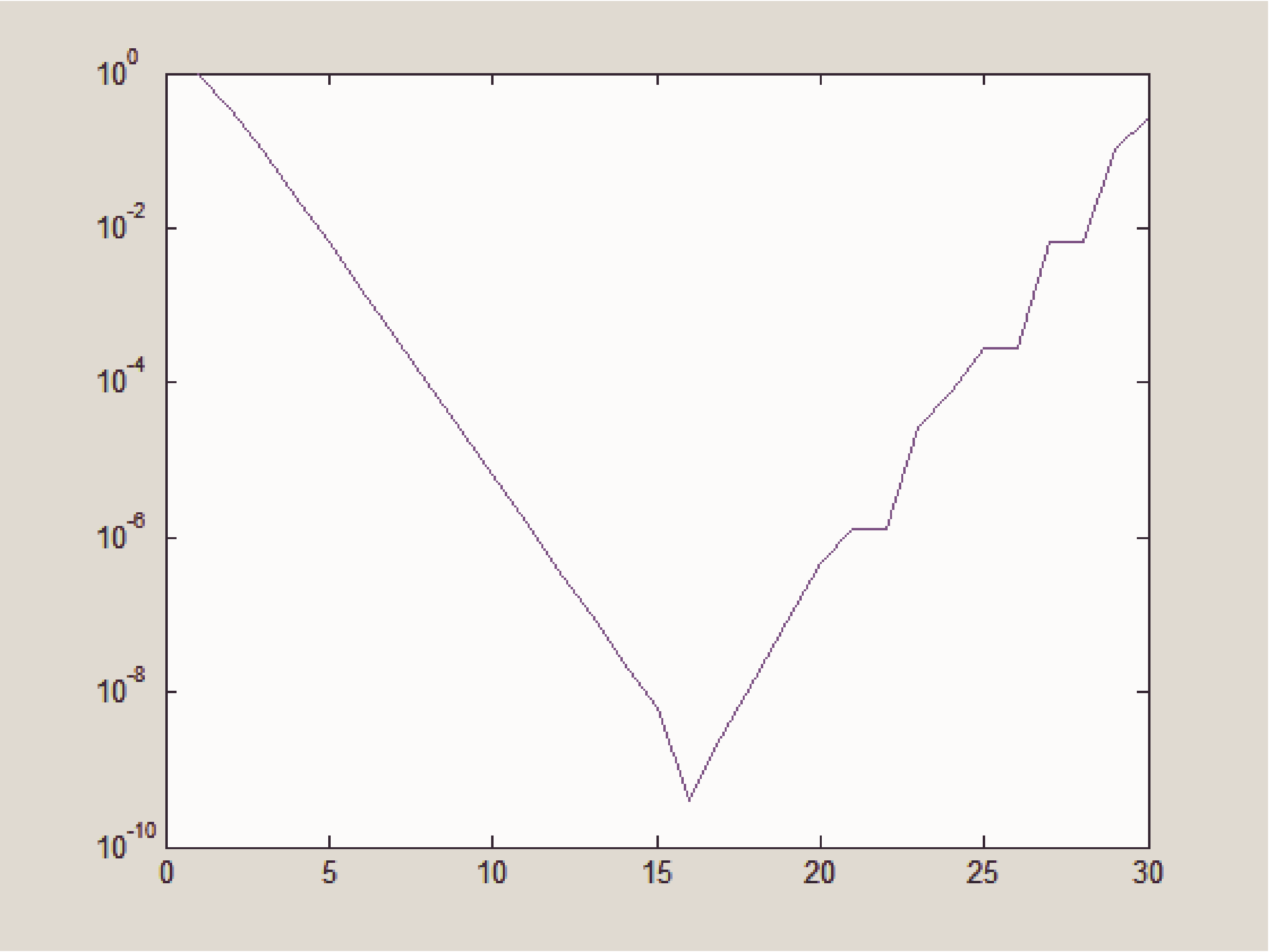

```
g = -1+x.*(7+x.*(-21+x.*(35+x.*(-35+x.*(21+x.*(-7+x))))));
plot(x,g)
```



h = (x-1).^7;
plot(x,h)



plot(x,h,x,g,x,f)




```
z(1) = 0; z(2) = 2;
for k = 2:29
    z(k+1) = 2^(k-1/2)*(1-(1-4^(1-k)*z(k)^2)^(1/2))^(1/2);
end
semilogy(1:30,abs(z-pi)/pi)
```

I.ARITHMETIC OPERATIONS and symbols

(1) format long, longe; format short, short e

(2) +, *, ^, ~, /, -

(3) suppress output: ending commands with “;”

(4) Complex:
real, imag, conj, i, j, angle, abs, compass

(5) Machine constants and special variables:
eps, realmin, realmax, pi

(6) loops: loop until condition

```
while (condition true)
end
```

Summary

- Roundoff and other errors
- Formats and floating point numbers
- Complex numbers

References

- Higham & Higham, Matlab Guide, SIAM
- SIAM News, 29(8), 10/98 (Arianne V failure)
- B5 Trailer; http://www.scifi.com/b5rangers/