



There are two different techniques whereby most computers today store the values of numbers: integer and floating-point. Integer storage has the nice property that it represents the exact value of the number stored; floating-point storage only guarantees a certain number of digits of precision. There is an upper limit to the values that can be stored in both integer and floating-point form. However, significantly larger numbers can be stored in floating-point storage than in integer storage.

By default, MATLAB sets the storage of numbers to double-precision floating-point representation. However, operations like reading images into MATLAB present the large volume of data in the more compact unsigned integer form.

Integers

Integers are represented in computer memory by blocks of data bits of various sizes. Memory is allocated in 8-bit increments, usually referred to as bytes; therefore, it is not surprising that integer storage comes in the same size increments. For a given size, the values of the data bits are represented in two different ways—signed or unsigned. Normally, of course, we expect a number to have both positive and negative values, and when the number of bits is large, this does not seem to have much impact. However, when a small number of bits are used to store a value, one of those bits must be used to show that the number is positive or negative. The range of numbers that can be stored is therefore reduced by 1 bit, a factor of 2. The following figure illustrates the internal storage of 8-bit unsigned and signed values.

Clearly, for 8 bits, the maximum value is 127 signed, or 255 unsigned. If this is not sufficient storage, numbers can be stored in 16-, 32-, or 64-bit words, with the corresponding increase in the maximum stored size.

Floating-Point Numbers

Floating-point numbers are stored in single precision (32 bits) or double precision (64 bits) using the IEEE 754 standard. As the name suggests, the storage format includes a mantissa and an exponent, each expressed

C-2

Appendix C Internal Number Representation

internally in a manner similar to integer storage. The fixed size of the mantissa leads to the fixed amount of precision of each storage type. The float data type gives 7 significant decimal digits; the double data type gives 15 significant decimal digits.¹

For details of these storage types, search the Web for “IEEE 754 standard.” At the time of writing, there was a good explanation at: http://www.geocities.com/SiliconValley/Pines/6639/docs/fp_summary.html

Parameters of Each Storage Type

The following table describes the most commonly used storage types available in MATLAB, their minimum and maximum values, and their equivalent names in C.

MATLAB Name	Size (Bytes)	Minimum Value	Maximum Value	C Name
uint8	1	0	255	unsigned char
int8	1	-128	127	char
uint16	2	0	65,536	unsigned short
int16	2	-32,768	32,767	short
uint32	4	0	4,294,967,295	unsigned int
int32	4	-2,147,483,648	2,147,483,647	int
float	4	~ -3.4E+38	~ 3.4E+38	float
double	8	~ -1.7E+308	~ 1.7E+308	Double

¹Note that although this seems to be a large amount of precision, you must always design your programs to preserve that precision. If, for example, you were to subtract two numbers almost equal in value, the precision of the result would be significantly worse than that of the original numbers.