liblat2eps v2.0

1 Introduction

liblat2eps is a small static library providing most of the functionality provided by the lat2eps command to user programs implemented in C or C++. It can be used for exporting lattice graphics in the Encapsulated PostScript (EPS) format directly from the programs, without generating an intermediary data file.

2 Installation

liblat2eps is open source software supplied in source code form. It can be built with the *make* command, which in addition to generating the `lat2eps` executable will also generate the `liblat2eps.a` static library. The *gcc* or *clang* compiler must be installed in the system for the build process to succeed. The `liblat2eps.a` file can be manually copied to a directory where the compiler searches for libraries, like, e.g., `/usr/local/lib`. The `lat2eps.h` header file declares the functions provided by the library and can be manually copied to a directory where the compiler searches for header files, like, e.g., `/usr/local/include`.

3 Library functions

liblat2eps provides the following functions:

```
int lat2eps_init(unsigned int width, unsigned int height)
```

The `lat2eps_init()` function initializes the resources required by the EPS generation functionality, including allocating memory for holding the lattice state until the EPS output can be generated. This function must be called before any other liblat2eps function is called. The *width* and *height* parameters, which can take values from 1 to 16384, define the maximum dimensions of the lattice. The function returns zero for indicating a failure, or a non-zero value for success.

```
void lat2eps_release()
```

The `lat2eps_release()` function releases the resources allocated by `lat2eps_init()`. It must be called after the EPS output is generated.

```
void lat2eps_set_site(unsigned int x, unsigned int y, int s)
```

The `lat2eps_set_site()` function sets the state of the site with coordinates x and y to value s, within the liblat2eps internal lattice state.

Typically, programs using liblat2eps should copy their own lattice state to the state held by liblat2eps by calling the `lat2eps_set_site()` function for each lattice site, before calling the `lat2eps_gen_eps()` function to actually generate the EPS output. Programs should avoid calling `lat2eps_set_site()` within their inner simulation loops, as it would likely cause a simulation slowdown. Instead, the programs should perform the copy only when the time for generating a graphic output has arrived, like after a determined number of Monte Carlo steps has been performed. Then the liblat2eps lattice state can be set with a loop like the following:

```
for (y = 0; y < L; ++y) {
        for (x = 0; x < L; ++x) {
                lat2eps_set_site(x, y, lat[y][x]);
        }
}
```

There are 256 different color indexes in liblat2eps, ranging from 0 to 255, which can be used to draw lattice sites of different values. Site values are mapped to color indexes according to a simple rule: *index* = *value* mod 256. Negative values are first converted to unsigned values (e.g., -1 is converted to 255, -2 to 254, and so on). Note: a palette of only 16 different colors is initially defined, repeated over the 256 possible indexes. Thus, if more than 16 colors are needed, they must be redefined through the lat2eps_set_color() function.

`int lat2eps_get_site(unsigned int x, unsigned int y)`

The `lat2eps_get_site()` function returns the value of the lattice site with coordinates given by x and y, previously set by a `lat2eps_set_site()` function call, or the default value 0, if no value has been set. This function has been defined for extension purposes and is not required by typical lattice graphic generation.

`void lat2eps_set_color(unsigned int index, unsigned int pal)`

The `lat2eps_set_color()` function sets the color index specified by the *index* parameter to a value defined by the *pal* parameter, which should be a numeric value in the 0xRRGGBB format (i.e., an integer composed by three bytes, where the most significant byte defines the red component for the color index, from 0x00 to 0xFF, the middle byte defines the green component, and the least significant byte defines the blue component). It must be called before the `lat2eps_gen_eps()` function.

`unsigned int lat2eps_get_color(unsigned int index)`

The `lat2eps_get_color()` function returns the palette definition associated with a color index specified by the *index* parameter. The returned value will be a numeric value in the 0xRRGGBB format.

`void lat2eps_text_out(float x, float y, float ax, float ay, float angle, unsigned int size, unsigned int color, const char *text)`

The `lat2eps_text_out()` function generates text entries over the lattice graphic. It can be used to generate text lines to appear in the graphic as they are, or tags that can be later replaced by LaTeX text using PSFrag. It must be called before the `lat2eps_gen_eps()` function. The parameters are the following:

- x is the horizontal coordinate where the text will be positioned. 0 is the leftmost coordinate, while the maximum horizontal coordinate is defined by the lattice width.
- y is the vertical coordinate where the text will be positioned. 0 is the topmost coordinate, while the maximum vertical coordinate is defined by the lattice height.
- *ax* defines the horizontal alignment. 0 for left-aligning the text relative to the X coordinate, 0.5 for centering it on the X coordinate, 1 for right-aligning it, etc.
- *ay* defines the vertical alignment. 0 for placing the top of the text on the Y coordinate, 0.5 for centering it on the Y coordinate, 1 for placing the bottom of the text on the Y coordinate, etc.
- *angle* defines the angle to rotate the text, in degrees (0 for horizontal).
- *size* defines the font size.
- *color* defines the color index used to draw the text.
- *text* is the text to be generated. Parentheses characters in the text must be escaped with backslashes.

`int lat2eps_gen_eps(const char *filename, unsigned int xoff, unsigned int yoff, unsigned int width, unsigned int height, unsigned int border, unsigned int scale)`

The `lat2eps_gen_eps()` function creates an EPS file containing a graphic representation of the lattice (or a region of the lattice), using the lattice state information previously set through `lat2eps_set_site()` calls, also generating text entries previously defined by `lat2eps_text_out()` calls. This function can be called one or more times, possibly for different regions, before the lattice state is released by a `lat2eps_release()` call. The function returns zero for indicating a failure, or a non-zero value for success. The parameters are the following:

- *filename* defines the name of the EPS file that will be created. If this parameter is `NULL`, then the EPS data will be written to the standard output.
- *xoff* defines the first lattice column that will be presented in the graphic.
- *yoff* defines the first lattice row that will be presented in the graphic.
- *width* defines the width (in sites) of the sublattice that will be presented in the graphic.
- *height* defines the height (in sites) of the sublattice that will be presented in the graphic.
- *border* defines the width of a black border generated around the lattice graphic, or 0 for generating a borderless graphic.
- *scale* defines the scale used in the conversion of the lattice data to EPS. E.g., when the scale is set to 3, each lattice site will generate a 3x3 pixel square in the EPS output.

4 Sample code

The source code presented below illustrates a simple usage of the functionality provided by liblat2eps. Providing that the `lat2eps.h` header and the `liblat2eps.a` library are installed to standard locations, it could be compiled and linked with the *gcc* compiler through the following command:

```
gcc sample.c -o sample -lm -llat2eps
```

```c
#include <math.h>
#include <lat2eps.h>

#define L   512

int main(int argc, char *argv[])
{
        int x, y;

        /* Initializes liblat2eps for a lattice of width L and height L. */
        if (!lat2eps_init(L, L)) return -1;

        /* Sets color number 9 to red:60 green:70 blue:a0 */
        lat2eps_set_color(9, 0x6070a0);

        /* Fills the lattice with a circular pattern. */
        for (y = 0; y < L; ++y) {
                for (x = 0; x < L; ++x) {
                        lat2eps_set_site(x, y, (int)sqrt(x*x+y*y)/50);
                }
        }

        /* Adds some text. */
        lat2eps_text_out(5, 5, 0, 0.5, -45, 15, 1, "Hello");
        lat2eps_text_out(5, L-30, 0, 1, 0, 25, 1, "TAG1");
        lat2eps_text_out(L-30, 10, 1, 1, 90, 25, 0, "TAG2");

        /* Generates an EPS file with the entire lattice, */
        /* with border width 1 and scale 1. */
        lat2eps_gen_eps("sample.eps", 0, 0, L, L, 1, 1);

        /* Releases resources. */
        lat2eps_release();

        return 0;
}
```

5 License

liblat2eps v2.0 is open source software copyrighted by André R. de la Rocha and licensed under the Apache License Version 2.0. A copy of the license can be obtained at: `http://www.apache.org/licenses/LICENSE-2.0`