Linguagem C para Engenharia

Instruções para o Trabalho Final

(ENG 03049)

Bardo E.J. Bodmann

(21 de maio de 2019)

Escola de Engenharia – EE
Universidade Federal do Rio Grande do Sul – UFRGS

Trabalho

Introdução

Um modelo simples para gerar um sistema crítico é o chamado modelo das pilhas de areia. A seguir o modeli bi-dimensional será introduzido. Em cada sítio sobre uma área quadrada há um valor que corresponde à inclinação da pilha. Esta inclinação acumula-se como grãos de areia que são colocados aleatoriamente sobre a pilha aumentando a sua inclinação, até que a inclinação excede um valor limiar específico, a altura em que esse sítio colapsa transferindo areia nos sítios adjacentes. Assim, a colocação aleatória de areia em um determinado sítio pode não ter efeito, ou pode causar uma reação em cascata que irá afetar todos os sítios na rede. Esta rede pode ser implementada por uma matriz cujos elementos contem o valor da inclinação.

Modelo

A tarefa a ser realizado até o final da disciplina visa a simulaç˜ao simples de avalanches seguindo o modelo introduzido por Bak,Tang e Wiesenfeld, também conhecido como modelo de pilhas de areia. Este modelo é um exemplo de um sistema dinâmico apresentando a propriedade de criticalidade auto-organizada. A ideia deste modelo é motivado pelo experimento que segue. Grãos de areia serão jogado de forma aleatória acima de uma mesa. Ao longo do tempo em certas posições há mais acumulo de areia que na vizinhaça ao redor até a configuração se torna instável e o morro começa desmoronar. Este processo termina quando uma nova confirguração estável (distribuição de areia acima da mesa) está sendo alcancado.

Para implementar este modelo num programa análogo a este experimento e numa forma simplificada, a mesa pode ser considerado uma estrutura discreta, ou seja em forma matricial, onde apenas existem sítios com coordenadas (x, y) cujos valores são inteiros.

Exemplo: A mesa têm cantos com a dimensão 100, então os valores para x e y podem assumir valores conforme $x, y \in [0, 100]$. O conteúdo do elemento da matriz representa o acúmulo de graus neste sítio. A matriz será inicializada com valores de zero. Repetitivamente, será escolhido de forma randômica um sítio que sera incrementado por um. Caso que o valor ultrapassa um limiar crítico, este desmorona e distribui uma unidade para todos os seus vizinhos que por sua vez podem desmoronar, ou seja, provocar uma avalanche.

As regras de iteração para o modelo são:

1. A configuração inicial é uma superfície plana $z(x, y) = 0$ para todos os x e y.

2. Utilize um gerador de números (pseudo-)aleatórios para determinar as coordenadas do sítio a ser incrementado. A função *rand* (abreviatura de random), definida na biblioteca *stdlib*, gera números aleatórios. Cada invocação da função produz um número aleatório no intervalo fechado $[0, ..., RAND\_MAX]$. A constante $RAND\_MAX$ está definida no arquivo-cabeçalho *stdlib.h*.

3. Para gerar números aleatórios r entre $0, ..., N$ tanto para x quanto para y utiliza-se a seguinte instrução.

$$r = N \frac{rand()}{RAND\_MAX}$$

4. Uma vez determinado o sítio, aplica-se o incremento do mesmo,

$$z(x, y) \to z(x, y) + 1 \qquad \text{para} \qquad z(x, y) < 4 \,, \tag{1}$$

$$\text{ou a regra da avalanche caso que } z(x, y) > 4, \tag{2}$$

$$z(x, y) \to z(x, y)4 \tag{3}$$

$$z(x \pm 1, y) \to z(x \pm 1, y) + 1 \tag{4}$$

$$z(x, y \pm 1) \to z(x, y \pm 1) + 1 \tag{5}$$

Recursos

Os seguntes recursos programatórias devem ser incluidos no programa:

- Entradas e saidas formatadas de arquivos.
- Estruturas de controle de fluxo.
- Ponteiros.
- Funções e ponteiros para funções.
- Alocação dinâmica de memoria.
- Estruturas.

Elaboração do trabalho

O programa a ser elaborado deve ser programado utilizando o padrão ANSI C. Junto ao código fonte do programa deve ser entrego um documento que descreve o próprio problema (uma extensão do primeiro paragrafo deste documento), os recursos de implementação e o método da "solução".

O documento deve incluir as instruções para manuseiar o programa, como entradas que devem ser fornecidos pelo usuário e as saidas fornecidas pelo programa. Como item final, o documento deve apresentar a lista de referências consultadas tais como revistas, livros, paginas no internet entre outras.

Entrega do tranalho

O trabalho deve conter o código fonte do programa, e a documentação, que deve ser entrego até o final do semestre conforme cronograma.

Apêndice

A utilização da função *rand*()

Esta função da biblioteca padrão *stdlib* gera números aleatórios. O seu protótipo é:

```
int rand(void);
```

Parameters: A função *rand*() retorna um número inteiro pseudo-aleatório. Este número inteiro é gerado no intervalo $[0, RAND\_MAX]$. O algorítmo de *rand*() utiliza uma inicialização, também chamado de semente, para gerar a seqüência de números pseudo-aleatórios. Por isto existe uma segunda função *srand*() que inicializa o gerador de números pseudo-aleatórios. Note, que a utilização de um número específico no argumento da função *srand*() produz sempre a mesma seqüência de números pseudo-aleatórios. A constante $RAND\_MAX$ é definedo na biblioteca padrão (*stdlib*).

Exemplo:

```
#include<stdio.h>
#include<stdlib.h>

int main ()
{
 int number, input;

   /* inicializar semente com um numero qualquer (aqui 314159) */
 srand(314159);

 /* Gere um numero aleatorio: */
 number = rand() % 10 + 1;
 do {
printf ("Guess the number (1 to 10): ");
scanf ("%d",&input);

if (numberinput)
printf ("The number is higher\n");

} while (number!=input);

printf ("Isto e verdadeiro!\n");

return 0;
}
```