



## Automatização dos cálculos de irradiação topográfica por meio de programação em linguagem C

Fernando Samuel Moreira Dias<sup>1</sup> Daniel Tesaro Cobo<sup>2</sup> Claudinei Rodrigues de Aguiar<sup>3</sup>  
Muriel de Souza Godoi<sup>4</sup>

23 novembro 2017

**Resumo** – O trabalho de representação topográfica de um terreno divide-se em duas partes: o levantamento em campo, para obtenção de dados, e a realização de cálculos, para a correta representação do terreno. Essa última parte requer muito tempo e esforço, além de estar sujeita a erros humanos, sendo assim, foi criado um programa em linguagem de programação C que realiza todos os cálculos necessários no processo de levantamento por irradiação topográfica, além de criar um arquivo que pode ser aberto no AutoCad com todos os pontos irradiados. O programa mostrou-se eficaz, e agora o próximo passo é adicionar outros processos de representação topográfica, como a planialtimetria e o nivelamento geométrico.

**Palavras-chave:** Topografia. Irradiação topográfica. Linguagem de Programação C. Code::Blocks.

### 1. INTRODUÇÃO

O homem sempre necessitou conhecer o meio em que vive, por questões de sobrevivência, orientação, segurança, guerras, navegação, construção, etc. No princípio a representação do espaço baseava-se na observação e descrição do meio. Com o tempo, surgiram técnicas e equipamentos de medição que facilitaram a obtenção de dados para posterior representação. A topografia foi uma das ferramentas utilizadas para realizar essas medições.

A palavra topografia tem sua origem no latim, *Topos* (lugar, região) *Graphen* (descrição), ou seja, descrição do terreno (VEIGA; ZANETTI; FAGGION, 2012). É um ramo da Geodésia geométrica cuja ciência tem

por objetivo conhecer, descrever e representar graficamente sobre uma superfície plana, partes da superfície terrestre, desconsiderando a curvatura do planeta Terra (COELHO JÚNIOR; ROLIM NETO; ANDRADE, 2014).

Além de medições de campo, a topografia compreende o cálculo de áreas, volumes e outras quantidades, assim como a preparação dos respectivos mapas e diagramas.

Atualmente o topógrafo utiliza instrumentos eletrônicos para medir, visualizar e registrar distâncias e posições de pontos automaticamente. Porém ainda são necessários muitos cálculos para obtenção de dados como coordenadas de pontos, e após isso

1 fernandosmdias@gmail.com, Universidade Tecnológica Federal do Paraná - UTFPR, Campus Apucarana, Apucarana, Paraná, Brasil.

2 danielcobo97@gmail.com, Universidade Tecnológica Federal do Paraná - UTFPR, Campus Apucarana, Apucarana, Paraná, Brasil.

3 rodriguesaguiar@utfpr.edu.br, Universidade Tecnológica Federal do Paraná - UTFPR, Câmpus Apucarana, Apucarana, Paraná, Brasil.

4 murielgodoi@gmail.com, Universidade Tecnológica Federal do Paraná - UTFPR, Câmpus Apucarana, Apucarana Paraná, Brasil.



normalmente é necessário representar todos esses dados em alguma ferramenta CAD (*Computer Aided Design*).

Para solucionar esse problema, foi desenvolvido um programa SACT (Sistema de Automação de Cálculos Topográficos) em linguagem C que realiza todos os cálculos envolvendo o método de levantamento topográfico taqueométrico por irradiação e cria um arquivo denominado "*script*", contendo a lista de todos os pontos irradiados e suas coordenadas cartesianas bidimensionais, que pode ser importado para o ambiente do *software* AutoCAD.

## 2. MATERIAIS E MÉTODOS

O desenvolvimento do trabalho foi baseado na utilização de dados reais de levantamento realizado em campo e a implementação do aplicativo utilizando a linguagem C, para tanto foi utilizada a estrutura disponibilizada pela UTFPR – Campus Apucarana e aplicativos com ambiente de programação. Dentro do exposto pode-se listar os seguintes itens utilizados:

- Tripé;
- Piquete;
- Régua estadimétrica;
- Nível de cantoneira;
- Trena;
- Teodolito Digital FOIF DT 402L, com precisão angular de 2"; e
- Ambiente de programação Code::Blocks, distribuído sob a licença GNU (*General Public License*).

O processo de levantamento topográfico taqueométrico por meio de irradiação é dividido em duas partes: o trabalho em campo e o processamento e cálculos topográficos.

O levantamento taqueométrico, também conhecido como estadimétrico, é uma técnica de medida indireta da distância horizontal entre dois pontos, utilizada em trabalhos topográficos. Nessa técnica, a partir do equipamento de medida (teodolito ou nível) instalado sobre um ponto, realiza-se a leitura dos fios

estadimétricos superior e inferior projetados sobre a mira estadimétrica instalada em um segundo ponto. A distância horizontal é calculada pela relação entre a leitura dos fios e a constante do equipamento (TULER; SARAIVA, 2014).

O desenvolvimento do trabalho em campo inicia-se com a escolha do ponto, sendo o mesmo materializado pela fixação de um piquete, servindo de referência para a instalação do teodolito, centragem, nivelamento e medição da altura do equipamento. Após isso, define-se um ponto como o "norte arbitrário", sendo esse o ponto referencial que materializa o eixo Y do sistema de coordenadas cartesianas locais e é utilizado até o fim do trabalho em campo. É recomendável o esboço de um croqui para melhor noção do espaço irradiado.

Após esses preparos, foram colocadas as estadias nos pontos determinados para a leitura dos fios médio, superior e inferior. As estadias são verticalizadas com o auxílio de um nível de cantoneira. Logo após a leitura dos fios, é anotado os ângulos horizontais e verticais obtidos com o teodolito em posição direta (limbo graduado orientado à esquerda do equipamento).

Após a leitura dos ângulos e dos fios estadimétricos para todos os pontos de interesse é iniciado o processo de utilização do programa SACT para realização dos cálculos desenvolvido nesse trabalho.

### 2.1 Cálculos e algoritmo computacional

O programa SACT foi criado em linguagem de programação C, utilizando o ambiente de programação Code::Blocks. C é uma linguagem de programação de propósito geral, estruturada e com controle de fluxo. Essa linguagem foi originalmente projetada e implementada no sistema operacional UNIX no DEC PDP-11, por Dennis Ritchie (KERNIGHAN; RITCHIE, 1988).

O usuário deve introduzir os seguintes dados:

- Número de pontos irradiados (máximo de 50 pontos)
- Valor do ângulo zenital ou zenital do ponto n



irradiado

- Valor da leitura do fio superior do ponto n
- Valor da leitura do fio inferior do ponto n
- Valor da leitura do fio médio do ponto n
- Especificar se o ângulo digitado é vertical ou zenital
- Valor do ângulo de horizontal ou de azimute do ponto n
- Coordenada no sistema local do ponto de estação inicial da irradiação

Com esses dados, é possível realizar todos os cálculos necessários para a determinação das coordenadas cartesianas locais de cada ponto e assim representar graficamente o local irradiado.

Para o controle de qualidade das leituras dos fios estadimétricos, utilizados para o cálculo da distância horizontal, é realizada uma análise da discrepância entre o fio médio ( $FM_{\text{observado}}$ ) lido pelo observador e o fio médio ( $FM_{\text{calculado}}$ ) calculado a partir da leitura dos fios superior (FS) e inferior (FI):

$$FM_{\text{calculado}} = \frac{FS + FI}{2} \quad (1)$$

sendo que as leituras dos fios estadimétricos são aceita se satisfazer a seguinte condição:

$$|FM_{\text{calculado}} - FM_{\text{observado}}| \leq 1 \text{ mm} \quad (2)$$

A análise para verificar se os valores lidos para os fios estadimétricos estão corretos é feita para cada ponto irradiado e inserido no aplicativo SACT.

Em seguida, a partir dos fios FS e FI e do ângulo zenital ( $z$ ) ou vertical ( $v$ ), são realizados os cálculos para a obtenção das distâncias horizontais ( $D_H$ ) entre o teodolito e os pontos. Os cálculos das distâncias são realizados pelas equações 3 ou 4, caso o usuário forneça os ângulos  $z$  ou  $v$ , respectivamente:

$$D_H = K (FS - FI) \text{sen}^2 (z) \quad (3)$$

$$D_H = K (FS - FI) \text{cos}^2 (v) \quad (4)$$

onde,  $K$  é a constante do equipamento, sendo igual a 100.

Com as distâncias horizontais obtidas, a próxima etapa é o cálculo das coordenadas relativas  $D_X$  e  $D_Y$  de cada ponto:

$$D_X = D_H \text{sen} (Az) \quad (5)$$

$$D_Y = D_H \text{cos} (Az) \quad (6)$$

onde  $Az$  é o azimute ou o ângulo horizontal entre o eixo  $Y$  e o alinhamento para o ponto considerado.

Após isso, é realizado o cálculo das coordenadas absolutas  $X$  e  $Y$  dos pontos irradiados, considerando que a coordenada da estação topográfica, de onde foram realizadas as irradiações, tem suas coordenadas conhecidas ou arbitradas pelo usuário. Assim, a partir das coordenadas  $X_0$  e  $Y_0$  da estação topográfica e das coordenadas parciais do  $i$ -ésimo ponto ( $D_{X_i}$  e  $D_{Y_i}$ ) é possível calcular as coordenadas absolutas ( $X_i$  e  $Y_i$ ) desse ponto:

$$X_i = X_0 + D_{X_i} \quad (7)$$

$$Y_i = Y_0 + D_{Y_i} \quad (8)$$

Se todos os dados inseridos estiverem corretos, o SACT irá calcular todas as distâncias horizontais e coordenadas dos pontos, exibindo uma tabela na tela com esses dados. Após isso, é criado um arquivo "script", que pode ser importado para o AutoCad, assim plotando todos os pontos calculados.

Esse arquivo é criado graças à um ponteiro do tipo "FILE \*file file=fopen", uma função que cria um arquivo do tipo .txt no local onde o programa está instalado. Se já houver um arquivo com o mesmo nome, ele irá sobrescrever o antigo.

Como na linguagem C não realiza cálculos com ângulos em graus, também necessária a conversão



para radianos utilizando a função “M\_PI”, que obtém todos os números de “pi” para realizar o cálculo mais precisamente possível, para isso foi desenvolvida uma função do tipo **void** especificamente para realizar o processo de conversão de graus para radianos.

Além disso, foram implementados algoritmos de controle de erro na inserção de dados, caso o usuário digite algum dado incorreto, como por exemplo ângulos e leitura dos fios da estadia. Caso o usuário digite algum ângulo ou fio de estadia incorreto, o programa irá mostrar uma mensagem de erro e solicitar novamente a digitação.

Para avaliar o SACT com dados reais, o mesmo foi utilizado em um levantamento topográfico realizado em um estacionamento próximo ao Bloco P da Universidade Tecnológica Federal do Paraná (UTFPR) – Campus Apucarana. Também foram realizados cálculos manuais para validação dos resultados obtidos.

### 3. RESULTADOS

Como resultado do desenvolvimento do SACT, obteve-se programa com código fonte em linguagem C, composto por 539 linhas de comando, contemplando o desenvolvimento de 9 funções do tipo **void**, onde foram implementados algoritmos para controle de erro no fluxo de dados do programa, conversões de unidades angulares, cálculos trigonométricos, menu de interface com o usuário, entre outros. Logo, obteve-se as seguintes funções que são utilizados pela função *main*:

- **void inicializacao\_utilidades():** submenu principal do menu de utilidades, onde solicita para que o usuário entre com os valores dos ângulos, faz verificação de inserção de valores inválidos e orienta o usuário sobre o formato correto de entrada dos valores;
- **void utilidades\_distancias():** solicita ao usuário a entrada dos valores dos fios estadimétricos para cada ponto e realiza o cálculo das distâncias horizontais, a serem utilizadas no cálculo das coordenadas

parciais;

- **void digitar\_azimutes\_irradiacao():** habilita o menu de interface para o usuário fornecer os valores dos azimutes necessários para os cálculos das coordenadas parciais dos pontos;
- **void radianos\_para\_graus():** converte o ângulo de radianos para graus, para visualização no final do menu de utilidades;
- **void graus\_para\_radianos():** converte o ângulo no formato graus, minutos e segundos em radianos, para ser possível os cálculos utilizando funções trigonométricas;
- **void converter\_angulos():** realiza a conversão dos ângulos de orientação azimute em rumo e vice-versa, com a análise de quadrantes;
- **void calcular\_coordenadas():** calcula as coordenadas absolutas dos pontos e escreve em um arquivo .txt;
- **void calculos\_irradiacao():** chama funções e executa a sequência de cálculos necessários para a irradiação;
- **void finalizacao\_irradiacao():** finaliza o processamento do projeto de irradiação apresentando em tela o resultado dos processamentos e informando sobre o arquivo gerado para importação em ambiente CAD.

A Figura 1 apresenta um fragmento do código desenvolvido, o qual realiza a solicitação para que o usuário entre com o ângulo do azimute.

```
void digitar_azimutes_irradiacao() {
    for(contador=0; contador<numero_pontos; contador++) {
        do {
            printf("Digite o valor do azimute do ponto %d:\n",contador+1);
            printf("Exemplo de digitacao: para 45 graus 37 minutos 19 segundos digite:\n");
            printf("45 37 19\n");
            scanf("%d %d %d",&graus_az[contador],&minutos_az[contador],&segundos_az[contador]);
            getchar();
            system("cls");
            if ((graus_az[contador]>359 || graus_az[contador]<0) ||
                (minutos_az[contador]>59 || minutos_az[contador]<0) ||
                (segundos_az[contador]>59 || segundos_az[contador]<0)) {
                printf("Angulo invalido,por favor,digite novamente...\n");
                printf("Graus devem estar entre 00 e 359!!\n");
                printf("Minutos e segundos devem estar entre 00 e 59!!\n");
                printf("Aberte ENTER para continuar...\n");
                fflush(stdin);
                getchar();
                system("cls");
            }
        } while ((graus_az[contador]>359 ||graus_az[contador]<0) ||
                (minutos_az[contador]>59 || minutos_az[contador]<0) ||
                (segundos_az[contador]>59 || segundos_az[contador]<0));
    }
}
//digita os azimutes necessarios para a irradiação
for (contador=0; contador<numero_pontos; contador++) {
```

Figura 1 – Exemplo do código fonte da função digitar\_azimutes\_irradiacao()



A Figura 2 apresenta a tela inicial do SACT.

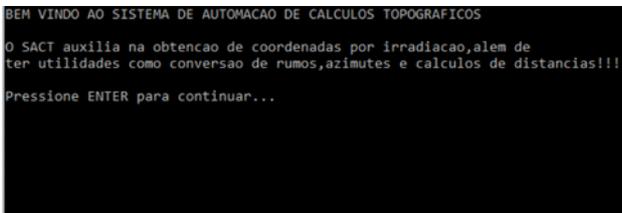


Figura 2 – Tela inicial do SACT

Após a entrada dos ângulos zenital (ou vertical), azimutes, fios estadimétricos e coordenadas do ponto estação, é realizado o processamento dos dados. Após a finalização dessa etapa é montada uma tabela mostrando as distancias e as coordenadas de cada ponto, bem como o aviso que o arquivo *script* está pronto e pode ser usado no AutoCad (Figura 3).

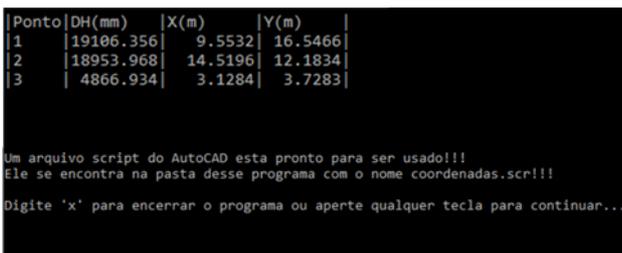


Figura 3 – Tela final após o processamento

Finalmente, a Figura 4 apresenta o resultado do levantamento por irradiação importado no AutoCad após o processamento pelo SACT.

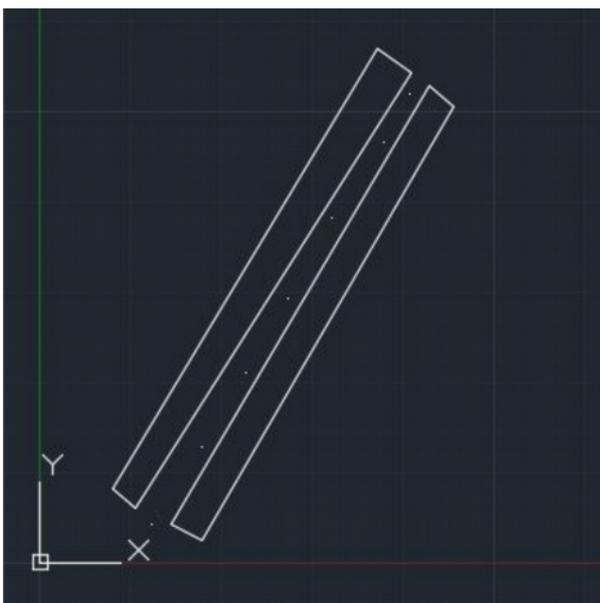


Figura 4 – Representação gráfica do estacionamento da UTFPR Campus Apucarana

Pelas análises dos resultados obtidos com o SACT em comparação àqueles obtidos pelos cálculos manuais, observou-se que não houve problemas de implementação no algoritmo do programa e os resultados obtidos mostraram que os cálculos foram realizados corretamente.

#### 4. CONCLUSÕES

Pode-se concluir que a utilização desse programa é altamente viável, por auxiliar e reduzir significativamente o tempo necessário para o processamento dos dados e a representação gráfica, considerando que basta apenas inserir os dados obtidos em campo. Além disso, é feito o *display* da tabela na tela contendo o resultado do processamento, bastando apenas que o usuário, caso deseje, tire uma *screenshot* para guardar os dados de cada ponto.

Com relação aos resultados obtidos, o SACT mostrou-se confiável apresentando resultados que foram validados por cálculos manuais.

Futuramente, serão implementados os cálculos para projetos utilizando outros métodos de levantamento topográfico, como planialtimetria por poligonização utilizando taqueométrica ou estação total, além do nivelamento geométrico. Porém, esses métodos são mais complexos, necessitando de mais variáveis e estrutura de código mais complexa, sendo a utilização de ponteiros e locação dinâmica de memória alguns dos recursos a serem utilizados.

#### AGRADECIMENTOS

Os autores agradecem à UTFPR pela disponibilização da infraestrutura. Ao Laboratório de Geodésia e Topografia do Curso de Engenharia Civil pela disponibilização dos equipamentos topográficos utilizados no levantamento de dados em campo.



## REFERÊNCIAS

COELHO JÚNIOR, José Machado; ROLIM NETO, Fernando Cartaxo ; ANDRADE, Julio da Silva Correa de Oliveira. **Topografia Geral**. Recife: EDUFRPE, 2014. 156 p.

KERNIGHAN, Brian W.; RITCHIE, Dennis M. **The C programming language**. 2. ed. New Jersey: Prentice Hall. 1988. 272p.

TULER, Marcelo; SARAIVA, Sergio. **Fundamentos de topografia**. Porto Alegre: Bookman, 2014. xv, 308 p.

VEIGA, Luis Augusto Koenig; ZANETTI, Maria Aparecida Zehnpfennig; FAGGION, Pedro Luis. **Fundamentos de topografia**. Curitiba: 2012. 274 p. Disponível em : <[http://www.cartografica.ufpr.br/docs/topo2/apos\\_topo.pdf](http://www.cartografica.ufpr.br/docs/topo2/apos_topo.pdf)>. Acesso em : 03/05/2017.



## Automation of calculations of topographic irradiation by programming in C language

Fernando Samuel Moreira Dias<sup>5</sup> Daniel Tescaro Cobo<sup>6</sup> Claudinei Rodrigues de Aguiar<sup>7</sup>  
Muriel de Souza Godoi<sup>8</sup>

23 novembro 2017

**Abstract** – Work of topographical representation of a terrain portion is carried out in two steps: the field survey, for obtaining data, and the accomplishment of calculations, for the correct representation of the terrain. This last part requires a lot of time and effort, besides being subject to human errors, thus, a programming language program C has been created that performs all the necessary calculations in the survey process by topographic irradiation method, besides creating a file that can be opened in AutoCad with all radiated points. The program proved to be effective, and the next step will be to add other topographical survey method, such as planialtimetry and geometric leveling.

**Keywords:** Topography. Topographic irradiation. C Programming Language. Code::Blocks.

**Correspondência:**

Claudinei Rodrigues de Aguiar

Rua Marçílio Dias, 635, CEP 86812-460, Apucarana, Paraná, Brasil.

Recebido: 30/10/2017

Aprovado: 23/11/2017

**Como citar:** DIAS, Fernando Samuel Moreira; et al. Automatização dos cálculos de irradiação topográfica por meio de programação em linguagem C. *Syn. scy. UTFPR*, Pato Branco, v. 12, n. 1, p. 247–253. 2017. ISSN 2316-4689 (Eletrônico). Disponível em: <<https://periodicos.utfpr.edu.br/synscopy>>. Acesso em: DD mmm. AAAA.

DOI: "registro apenas quando a revista for depositada no portal do PERI"

Direito autoral: Este artigo está licenciado sob os termos da Licença **Creative Commons** Atribuição 4.0 Internacional.

<sup>5</sup> fernandosmdias@gmail.com, Universidade Tecnológica Federal do Paraná - UTFPR, Campus Apucarana, Apucarana, Paraná, Brasil.

<sup>6</sup> danielcobo97@gmail.com, Universidade Tecnológica Federal do Paraná - UTFPR, Campus Apucarana, Apucarana, Paraná, Brasil.

<sup>7</sup> rodriguesaguiar@utfpr.edu.br, Universidade Tecnológica Federal do Paraná - UTFPR, Câmpus Apucarana, Apucarana, Paraná, Brasil.

<sup>8</sup> murielgodoi@gmail.com, Universidade Tecnológica Federal do Paraná - UTFPR, Câmpus Apucarana, Apucarana Paraná, Brasil.