UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

# TOMOGRAFIA ÓPTICA DIFUSA: *HARDWARE* E TESTES *IN VITRO*

Marcelo Marques Nogueira

São Paulo
2007

UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

# TOMOGRAFIA ÓPTICA DIFUSA: *HARDWARE* E TESTES *IN VITRO*

Trabalho de formatura apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do título de Graduação em Engenharia

Marcelo Marques Nogueira

Orientador: Prof. Dr. Raul Gonzalez Lima

Área de concentração:
Engenharia Mecânica

São Paulo

2007

## FICHA CATALOGRÁFICA

**Nogueira, Marcelo Marques**
**Tomografia óptica difusa: *hardware* e testes *in vitro* / M.M. Nogueira. – São Paulo, 2007.**
**83 p.**

**Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecânica.**

**1.Tomografia 2.Método dos elementos finitos 3.Imageamento (Bioengenharia) 4.Equação de Helmholtz I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecânica II.t.**

# AGRADECIMENTOS

Ao professor Dr. Raul Gonzalez Lima, com o qual eu aprendi tudo que este Trabalho retrata, e muito mais. Com ele, aprendi a atacar problemas inéditos com serenidade, persistência, rigor e criatividade.

Ao amigo Fernando Silva de Moura, que em diversas ocasiões me auxiliou a compreender pontos da teoria de tomografia aparentemente indecifráveis, sempre com a elegância de quem sabe exatamente do que está falando.

Aos meus pais, que sempre me proporcionaram subsídio material, emocional e espiritual em todos os meus projetos. Mais que isso, agradeço pelos valores que me transmitiram ao longo de toda minha vida, os quais me tornaram a pessoa que sou hoje.

“*Success is the ability to go from one failure to another with no loss of enthusiasm.*”

Sir Winston Churchill (1874 - 1965)

# RESUMO

O presente Trabalho estudou um tema de pesquisa em bioengenharia: a tomografia óptica difusa. A obtenção de imagens tomográficas é uma atividade de importância inquestionável na medicina, o que dá fundamento ao desenvolvimento de técnicas tomográficas como a tomografia computadorizada e a tomografia por impedância elétrica. Com este Trabalho, procurou-se atacar o problema de geração de imagem de uma nova maneira: através do uso de luz. O baixo poder de penetração da mesma em tecidos humanos permite que apenas imagens superficiais possam ser obtidas. Entretanto, essa técnica fornece uma vantagem única: a capacidade de fornecer informação funcional sobre o tecido estudado, como nível de oxigenação. Para o estudo dessa técnica não ortodoxa de tomografia, construiu-se um protótipo de tomógrafo de primeira geração, para uso em laboratório. O protótipo consiste em uma matriz circular de emissores e receptores de raios infravermelhos, montada ao redor de uma cuba de plástico. A mesma pode ser preenchida com água turva, e um objeto estranho e opaco é colocado lá dentro. O objetivo do protótipo é identificar a posição e formato deste objeto. Para esse fim, programou-se um *software* para gerar imagens a partir das medidas do protótipo. O programa emprega o algoritmo Caixa-Preta para tanto, e algumas imagens experimentais foram obtidas. Espera-se que este seja o primeiro de uma série de aparatos que eventualmente culminará em uma versão com aplicação clínica.

# ABSTRACT

The purpose of this project has been the study of a research branch within bioengineering: diffuse optical tomography. The relevance of acquiring tomographical images is beyond question in the medical scope. This reason alone suffices to explain the development of tomographic techniques, such as computadorized tomography (CAT-scan) and electrical impedance tomography. In this project, one attempts to attack this image-generating problem in an innovative way: by means of the use of light. Its low penetration capability in human tissue restricts the possibility of image acquiring to superficial imagery. Nonetheless, this technique has a unique advantage: it provides the physician with functional data on the tissue under study, such as oxygenation level. In order to study this non-orthodox tomography technique, a first generation prototype has been built (for sole laboratory usage). This prototype consists of a circular matrix of infrared ray emitters and receptors, which is mounted onto a plastic bowl. It may be filled with turbid water, and an anomalous, opaque object may be placed in it. The purpose of the prototype is to identify this object's position and shape. On these grounds, a piece of software has been programmed, so as to generate images from the prototype's measurements. The program employs the Black Box algorithm, and a few experimental images have been acquired. Hopefully, this shall be the first of a series of devices which will eventually reach the level of sophistication necessary for clinical application.

# LISTA DE ILUSTRAÇÕES

# LISTA DE TABELAS

# SUMÁRIO

# 1. INTRODUÇÃO

## 1.1 Tema e Objetivos do Trabalho

Este Trabalho de Formatura em Engenharia Mecânica tem como tema a tomografia óptica difusa, que pertence ao campo da Bioengenharia. A tomografia é o processo de geração de imagens de objetos localizados em um plano. Em aplicações clínicas, essa definição geral pode ser restrita à obtenção de imagens de órgãos do corpo que estão em certo plano. A tomografia óptica é um tipo de tomografia que emprega a luz como meio para geração dessas imagens. Ela se contrapõe, por exemplo, à tomografia por impedância elétrica (que emprega diferenças de potencial) e à tomografia computadorizada comum (que emprega raios-X). Finalmente, a tomografia óptica difusa ressalta o fato que a luz sofre difusão (além de absorção) quando penetra em tecidos humanos, em contraposição aos raios-X, que os atravessam em linha reta.

Este Trabalho tem dois objetivos principais. Primeiro, a construção de um protótipo funcional de tomógrafo óptico de primeira geração, para uso em laboratório. Segundo, o emprego desse aparato para a obtenção de imagens de um meio no qual uma anomalia é inserida. Mais precisamente, para a determinação da posição e forma de um corpo estranho presente em um meio homogêneo.

## 1.2 Relevância e Justificativa da Escolha do Tema

Comparado aos raios-X e à corrente elétrica, a luz possui um poder de penetração bastante reduzido em tecidos humanos. Por conta disso, o tomógrafo óptico é primordialmente usado para se obter imagens superficiais desses tecidos. Apesar dessa limitação, as aplicações médicas potenciais do mesmo são várias, entre as quais se podem enumerar duas:

Avaliação da condição de vascularização periférica em portadores de diabetes. Uma das características dessa doença é a redução da vascularização das extremidades de seus portadores. Isso pode levar à falta de oxigenação das células, necroses, e até mesmo obrigar a amputação de extremidades. Através da tomografia óptica, é possível controlar a vascularização dessas extremidades, e monitorar se ela está ficando comprometida. Em caso positivo, o diagnóstico permite a adoção de medidas precoces, as quais visam à redução da gravidade do problema.

Localização de artérias principais de irrigação da pele. Um procedimento comum em cirurgia plástica consiste no transplante de uma porção de pele saudável para uma região adjacente comprometida. O exemplo clássico desse caso é um traumatismo, por exemplo, no calcanhar, devido ao qual o tecido epitelial foi destruído. No procedimento padrão em uma situação como essa, os cirurgiões plásticos removem uma porção saudável da pele da batata da perna para cobrir o tecido subcutâneo exposto na região do calcanhar. A escolha da localização e dimensão da porção de pele a ser removida da batata da perna é função da localização do vaso principal que a irriga. Tal vaso está disposto no sentido radial (em relação à perna), e vem de perto do osso até a pele, onde se subdivide em arteríolas. O problema enfrentado pelos cirurgiões é a determinação precisa da posição dessa artéria. Vale mencionar que a determinação errônea da localização da artéria acarreta a morte precoce do tecido transplantado, o qual perde, nesse caso, sua função de proteção. Os métodos usados atualmente para tanto se mostram ineficientes e caros, ao passo que a tomografia óptica se apresenta como forte candidata a realização dessa tarefa.

Além dessas duas aplicações, vale mencionar uma vantagem intrínseca que a tomografia óptica possui em relação aos outros métodos: a absorção da luz pelo oxigênio. Como conseqüência desse fenômeno físico, é possível determinar o nível de oxigenação dos tecidos analisados através da tomografia óptica, devido à sua maior ou menor absorção de luz. Em outras palavras, a tomografia óptica é capaz de fornecer inclusive informação funcional sobre o tecido analisado.

Tendo em mente todas as aplicações médicas vislumbradas aqui, justifica-se a escolha do tema desse Trabalho.

## 1.3 **Escopo do Trabalho e Métodos Empregados**

Conforme mencionado, a utilização final de um tomógrafo óptico jaz na área médica, onde ele pode ser empregado para se obter imagens superficiais de tecidos humanos. Entretanto, este Trabalho se restringirá à construção de um protótipo para uso laboratorial, e não clínico. O motivo dessa escolha é a grande complexidade da tecnologia de tomografia óptica, o que impede que a mesma seja completamente dominada no espaço de tempo disponível para a realização deste Trabalho. E a intenção dessa escolha é que este protótipo seja o primeiro de uma série, a qual eventualmente atingirá o nível de sofisticação necessário à aplicação clínica.

O protótipo será composto por agentes emissores de luz (LED's) e agentes receptores de luz (fotodiodos). Cada conjunto de agentes estará disposto em matrizes circulares de seis unidades cada. A geometria dessas matrizes, assim como a posição relativa entre elas será projetada e, obviamente, conhecida. Através da emissão de luz, controlada uma a uma, e das leituras dos agentes receptores de luz, um *software* deverá ser capaz de construir a imagem tomográfica.

Uma vez pronto, o protótipo deverá ser empregado sob as condições controladas que um laboratório fornece. A princípio se utilizará uma cuba com água e leite (para tornar a solução translúcida) e uma esfera negra nela mergulhada, a qual assumirá o papel de um corpo estranho ao meio. Com o protótipo e o *software* se tentará precisar a posição e o formato desse corpo estranho.

# 2. REVISÃO BIBLIOGRÁFICA E RESUMO TEÓRICO

O texto que segue é um resumo de um trabalho [1] realizado no campo de tomografia óptica difusa, e que guarda semelhanças com aquele que se propõe neste trabalho. Em seguida, discute-se equação física que explica o fenômeno de difusão. A mesma é tratada de forma a se tornar aplicável para o segundo objetivo desde trabalho: a geração da imagem tomográfica a partir das leituras nos fotodiodos.

## 2.1 Um Experimento de Tomografia Óptica Difusa

Para o experimento em questão, foram utilizados 25 emissores de luz na base de um cubo de 8x8x6cm, e 25 receptores em seu topo, conforme a Ilustração 1. O cubo é preenchido por um meio altamente espalhador de luz, o qual não foi especificado no artigo. Além desse meio homogêneo, duas anomalias esféricas foram colocadas no cubo, também mostradas na Ilustração 1. Uma delas absorve fortemente a luz, ao passo que a outra a espalha intensamente. É justamente nessas anomalias que está o interesse dos pesquisadores. Ativou-se cada emissor de luz separadamente e realizou-se a leitura em cada um dos receptores de luz.

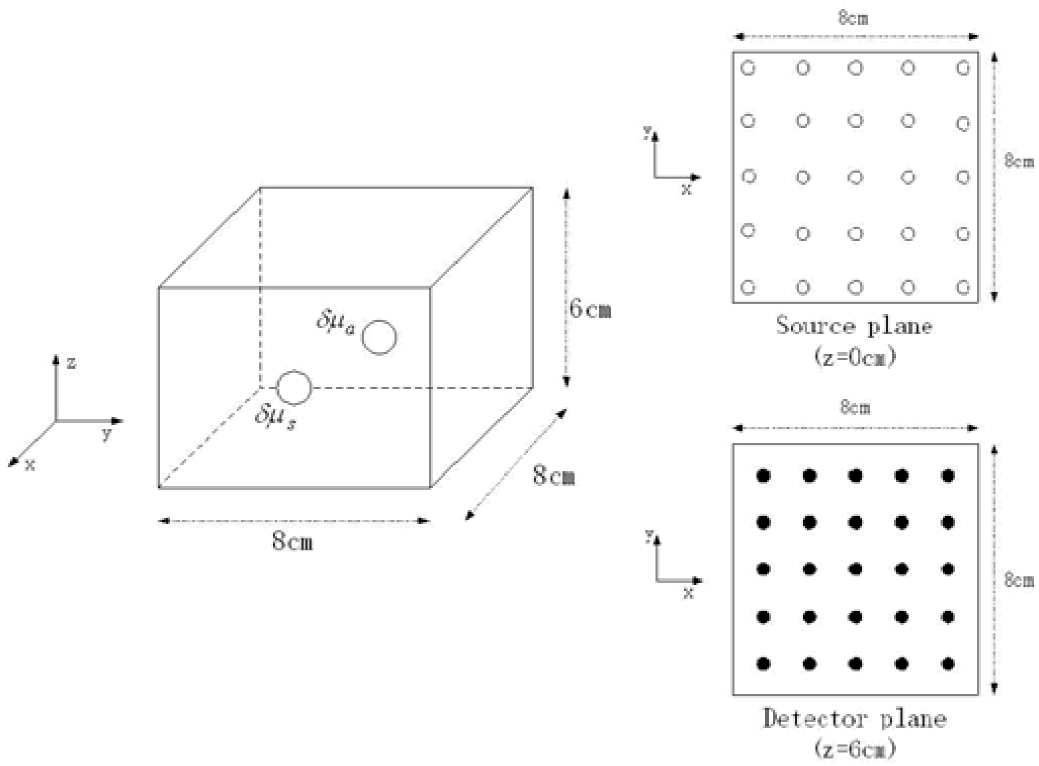



**Ilustração 1: Esboço da simulação realizada em [1].**

Através da análise dos dados colhidos, foi realizado um processamento de dados, cujo embasamento teórico é explicado adiante nesse capítulo. O objetivo do mesmo é a construção de imagens tomográficas em vários planos desse cubo. Isto é, para vários planos eqüidistantes entre a base e o topo do cubo, traçou-se uma imagem que mostra a localização de anomalias no meio homogêneo. A Ilustração 2 mostra os resultados finais da experiência. Cada uma das imagens corresponde a um desses planos eqüidistantes, conforme indicado pelo valor de “z” no topo das mesmas. As regiões em vermelho indicam a presença de anomalias ao meio homogêneo, o qual é representado pelas regiões em azul escuro.

As 12 imagens acima mostram a localização de anomalias de absorção de luz, ao passo que as 12 imagens abaixo mostram a localização de anomalias de espalhamento de luz.

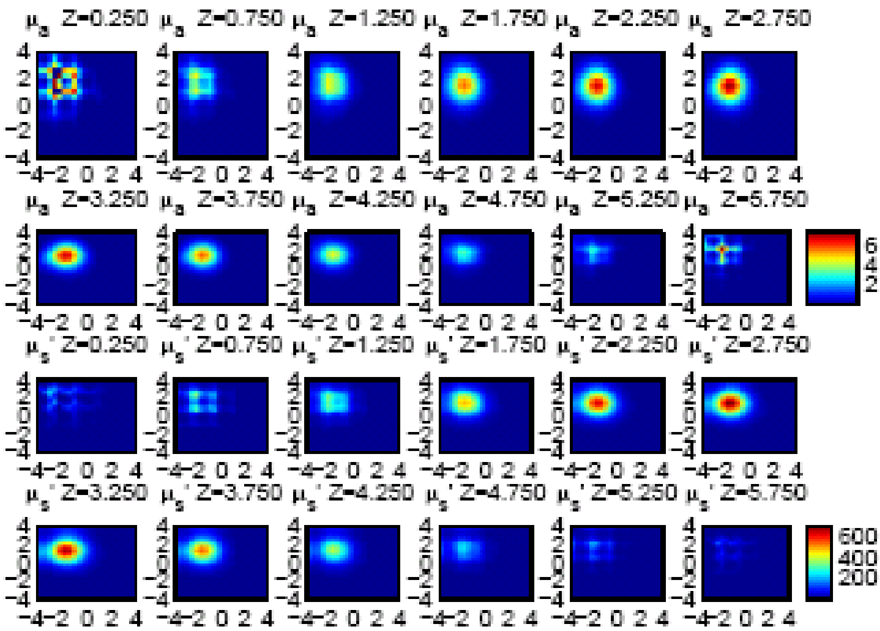



**Ilustração 2: Resultado do experimento de CAO.**

Como se pode ver, o resultado da experiência é um sucesso, uma vez que ambas as anomalias (a de absorção e a de espalhamento) foram localizadas independentemente.

## 2.2 **Embasamento Teórico: a equação de Boltzmann**

O problema da tomografia óptica difusa é definido em um domínio convexo $\Omega \in \Re^3$, que contém o objeto no qual se deseja realizar a tomografia. Tanto os emissores (LED's) quanto os receptores (fotodiodos) encontram-se na fronteira $\partial\Omega$ dessa região. As anomalias que se deseja detectar encontram-se no interior dessa região.

Quando o LED emite raios infravermelhos, fótons penetram na região $\Omega$. Eles são descritos pelo campo $\phi(r,\theta,t)$, que representa a densidade de fótons no ponto $r \in \Omega$ do domínio, no instante $t$ de tempo, e com velocidade na direção $\theta$:

- A unidade de $\phi$ é $m^{-3}$.
- $\theta = [sen(\alpha)\cdot\cos(\beta) \quad sen(\alpha)\cdot sen(\beta) \quad \cos(\alpha)]^T$ é uma direção definida por $\alpha$, ângulo de $\theta$ projetado em $xy$, e por $\beta$, ângulo de $\theta$ perpendicular a $xy$.

A partir dessa grandeza, determina-se a densidade de fótons $\Phi$ para cada ponto $r$ do domínio e instante $t$ de tempo. Para tanto, basta integrar $\phi(r,\theta,t)$ em todas as direções $\theta$ possíveis (representadas pelo conjunto $S^2$):

$$\Phi(r,t) = \int_{S^2} \phi(r,\theta,t)d\theta \tag{2.1}$$

É com base nessa densidade de fótons $\Phi$ que se enuncia a equação de transporte de Boltzmann, a qual descreve o fenômeno físico da propagação de fótons um meio homogêneo. A aproximação de difusão dessa equação é dada por [2] e [6]:

$$-\nabla\cdot[D\nabla(\Phi)] + \mu_a\Phi + \frac{1}{c}\frac{\partial\Phi}{\partial t} = q_0 \tag{2.2}$$

Nessa expressão:

- "$D = D(r)$" é o coeficiente de difusão $(m)$, definido por: $D = \dfrac{1}{3\cdot(\mu_a + \mu_s')}$.
- "$r$" é o vetor posição $(m,m,m)$.

- "$c$" é a velocidade da luz no meio $\left(m/s\right)$.
- "$\mu_a(r)$" é o coeficiente de absorção $\left(m^{-1}\right)$.
- "$\mu_s'(r)$" é o coeficiente de espalhamento reduzido $\left(m^{-1}\right)$, definido por: $\mu_s' = \mu_s \cdot (1-g)$.
- "$\mu_s(r)$" é o coeficiente de espalhamento $\left(m^{-1}\right)$.
- "$g$" é o valor esperado do co-seno do ângulo de espalhamento.
- "$q_0 = q_0(r,t)$" é uma fonte de fótons $\left(m^{-4}\right)$.

Embora essa equação descreva o fenômeno físico, ela é inadequada ao tratamento pelo método dos elementos finitos. De fato, [2] recomenda que ela seja transformada em uma equação de Helmholtz. Isso é feito em duas etapas: um artifício matemático seguido de uma mudança de variáveis.

O artifício matemático consiste em reescrever a equação assim:

$$-\nabla \cdot \left[\sqrt{D}\sqrt{D}\nabla(\Phi)\right] + \mu_a\Phi + \frac{1}{c}\frac{\partial\Phi}{\partial t} = q_0 \tag{2.3}$$

$$-\nabla\left(\sqrt{D}\right)\cdot\sqrt{D}\nabla(\Phi) - \sqrt{D}\nabla\cdot\left(\sqrt{D}\nabla(\Phi)\right) + \mu_a\Phi + \frac{1}{c}\frac{\partial\Phi}{\partial t} = q_0 \tag{2.4}$$

$$-\nabla\left(\sqrt{D}\right)\cdot\sqrt{D}\nabla(\Phi) - \sqrt{D}\left\{\nabla\left(\sqrt{D}\right)\cdot\nabla(\Phi) + \sqrt{D}\nabla^2(\Phi)\right\} + \mu_a\Phi + \frac{1}{c}\frac{\partial\Phi}{\partial t} = q_0 \tag{2.5}$$

$$-2\sqrt{D}\nabla\left(\sqrt{D}\right)\cdot\nabla(\Phi) - D\nabla^2(\Phi) + \mu_a\Phi + \frac{1}{c}\frac{\partial\Phi}{\partial t} = q_0 \tag{2.6}$$

$$-2\nabla\left(\sqrt{D}\right)\cdot\nabla(\Phi) - \sqrt{D}\nabla^2(\Phi) + \frac{\mu_a}{\sqrt{D}}\Phi + \frac{1}{c\sqrt{D}}\frac{\partial\Phi}{\partial t} = \frac{q_0}{\sqrt{D}} \tag{2.7}$$

A mudança de variável é a seguinte:

$$U = \sqrt{D}\Phi \tag{2.8}$$

Calculando o laplaciano, segue que:

$$\nabla^2(U) = \nabla^2\left(\sqrt{D}\Phi\right) \tag{2.9}$$

$$\nabla^2(U)=\nabla^2\left(\sqrt{D}\right)\Phi+\sqrt{D}\nabla^2(\Phi)+2\nabla\left(\sqrt{D}\right)\cdot\nabla(\Phi) \qquad \textbf{(2.10)}$$

$$-2\nabla\left(\sqrt{D}\right)\cdot\nabla(\Phi)-\sqrt{D}\nabla^2(\Phi)=\nabla^2\left(\sqrt{D}\right)\Phi-\nabla^2(U) \qquad \textbf{(2.11)}$$

Aplicando as equações acima, segue que:

$$\nabla^2\left(\sqrt{D}\right)\Phi-\nabla^2(U)+\frac{\mu_a}{\sqrt{D}}\Phi+\frac{1}{c\sqrt{D}}\frac{\partial\Phi}{\partial t}=\frac{q_0}{\sqrt{D}} \qquad \textbf{(2.12)}$$

$$-\nabla^2(U)+\left(\frac{\mu_a}{\sqrt{D}}+\nabla^2\left(\sqrt{D}\right)+\frac{1}{c\sqrt{D}}\frac{\partial}{\partial t}\right)\Phi=\frac{q_0}{\sqrt{D}} \qquad \textbf{(2.13)}$$

$$-\nabla^2(U)+\left(\frac{\mu_a}{\sqrt{D}}+\nabla^2\left(\sqrt{D}\right)+\frac{1}{c\sqrt{D}}\frac{\partial}{\partial t}\right)\frac{U}{\sqrt{D}}=\frac{q_0}{\sqrt{D}} \qquad \textbf{(2.14)}$$

$$-\nabla^2(U)+\left(\frac{\mu_a}{D}+\frac{\nabla^2\left(\sqrt{D}\right)}{\sqrt{D}}+\frac{1}{cD}\frac{\partial}{\partial t}\right)U=\frac{q_0}{\sqrt{D}} \qquad \textbf{(2.15)}$$

Define-se, finalmente, o termo $\eta=\eta(r)=\frac{\mu_a}{D}+\frac{\nabla^2\left(\sqrt{D}\right)}{\sqrt{D}}+\frac{1}{cD}\frac{\partial}{\partial t}$, e a equação torna-se:

$$-\nabla^2(U)+\eta U=\frac{q_0}{\sqrt{D}} \qquad \textbf{(2.16)}$$

Essa equação é uma equação de Helmholtz não-homogênea. Seu tratamento matemático é mais fácil que o tratamento da equação de difusão de Boltzmann, enunciada inicialmente. É essa a equação que será empregada na solução do problema através do método dos elementos finitos.

# 3. *HARDWARE*: MATERIAIS E MÉTODOS

Conforme previsto no cronograma (vide ANEXO C e ANEXO D), foi dado foco à construção do protótipo (*hardware*) e seu controle durante o primeiro semestre de 2007. Neste capítulo, o resultado desta etapa do trabalho será detalhado. Isto é, os pormenores da construção do protótipo serão explicados.

O protótipo é composto por três componentes principais: a matriz de emissores e receptores de luz, a placa de aquisição de dados, e o circuito elétrico entre eles. Além disso, para efetuar o controle de emissão de luz e as leituras nos receptores, é necessário um *software* que controle o mesmo, o qual também é explicado aqui.

## 3.1 **Matriz de Emissores e Receptores**

Essa matriz foi montada sobre um recipiente cilíndrico de plástico, de 10cm de diâmetro e 20cm de altura, como mostra a Ilustração 3:

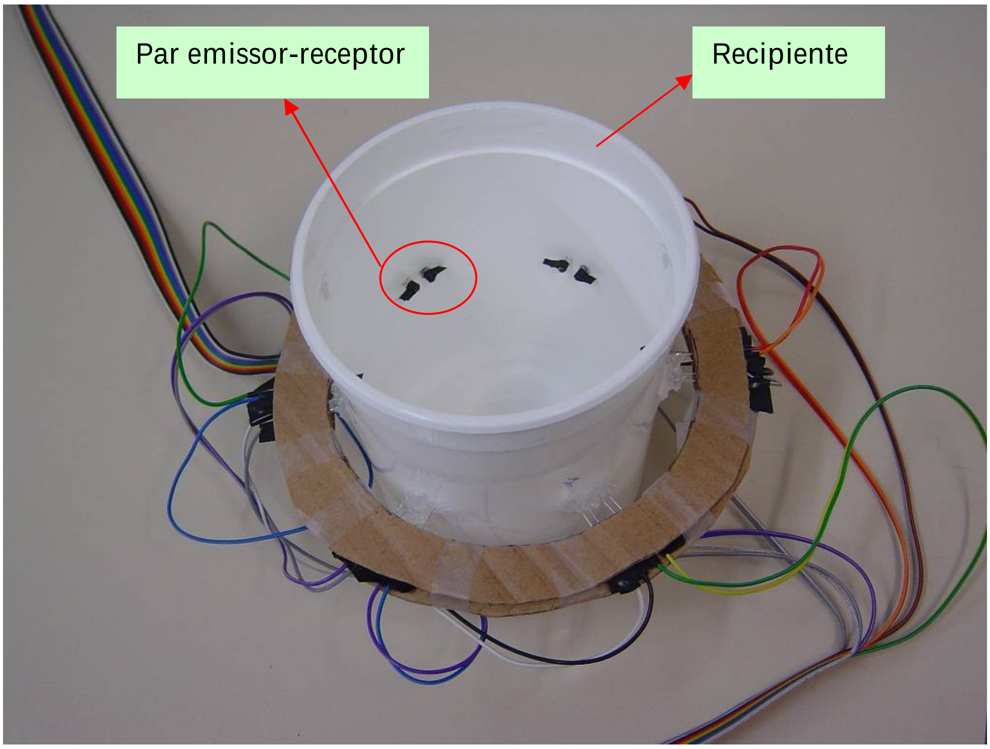



**Ilustração 3: Visão geral da matriz de emissores e receptores.**

A matriz de emissores é composta por seis LED's infravermelhos, dispostos uniformemente ao longo da circunferência do recipiente. Semelhantemente, a matriz de receptores é composta por seis fotodiodos sensíveis a infravermelho, também dispostos equidistantemente ao longo da circunferência do recipiente. O desvio angular entre as matrizes é tão pequeno quanto foi possível manufaturar. Isto é: ao lado de cada LED existe um fotodiodo, a uma distância de aproximadamente 1cm. O motivo dessa escolha é garantir que sempre existe um fotodiodo diametralmente oposto a cada LED. Dessa forma, quando acionado, um LED projeta seus raios infravermelhos normalmente sobre um único fotodiodo. A posição relativa entre os componentes pode ser mais bem vista na Ilustração 4:

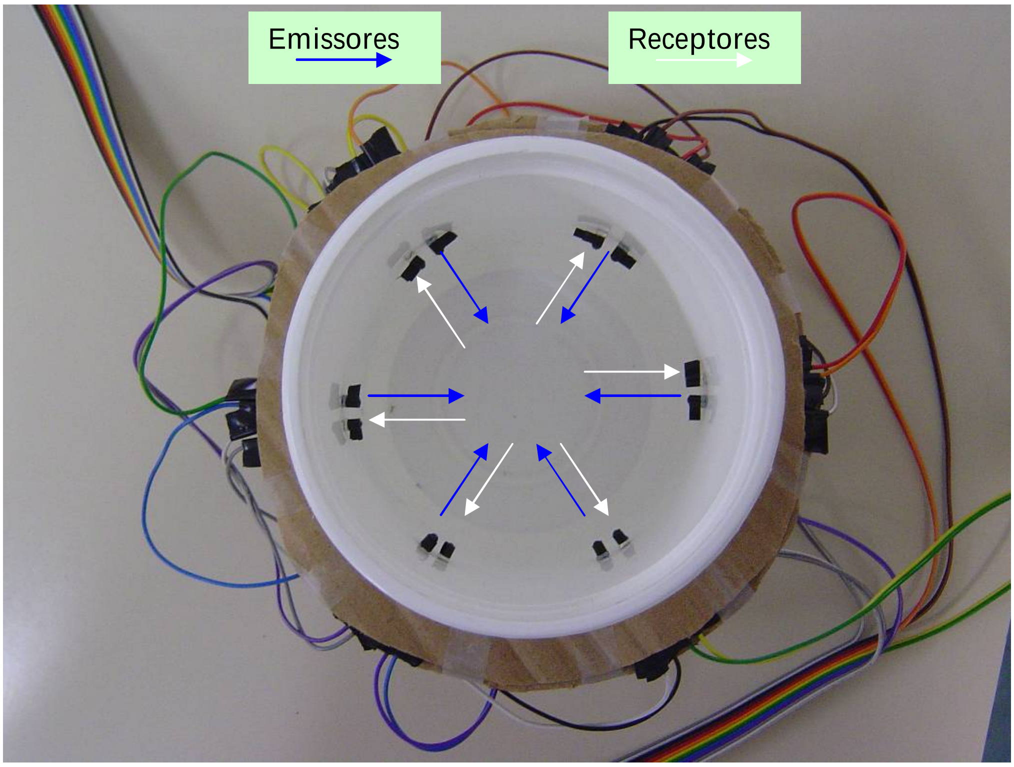



**Ilustração 4: Vista superior da matriz de emissores e receptores.**

Os LED's e os fotodiodos estão envoltos por um "espaguete" preto. Esse componente consiste em um tubo de plástico, cujo diâmetro pode ser reduzido através do aquecimento. O objetivo dessa medida é, por um lado, garantir que os raios que deixam os LED's sejam normais ao mesmo, e por outro, impedir que raios incidentes fora da lente não alterem a leitura dos fotodiodos. A Ilustração 5 mostra, no detalhe, esse isolamento.

**Ilustração 5: Detalhe do isolamento dos emissores e receptores.**

Optou-se pelo uso de LED's e fotodiodos infravermelhos por causa da máxima excitação que aqueles provocam sobre estes. No mercado, existem LED's que emitem diferentes comprimentos de onda, assim como fotodiodos sensíveis a vários comprimentos de onda. O problema é que, fora do comprimento de onda nominal, o LED emite uma intensidade de luz pequena, e o fotodiodo, por sua vez, apresenta problemas de detecção. Sendo assim, optou-se pelo uso de um par já pronto: o emissor e receptor de infravermelho, comumente usado em equipamentos eletrônicos como controles remotos.

Para que se pudesse encher o recipiente com água, vedou-se a região dos componentes ópticos. E para garantir que os mesmos ficassem alinhados segundo um mesmo plano, empregou-se um par de discos de papelão, que mantém as pernas dos LED's e dos fotodiodos orientadas. A Ilustração 6 mostra bem esses detalhes:

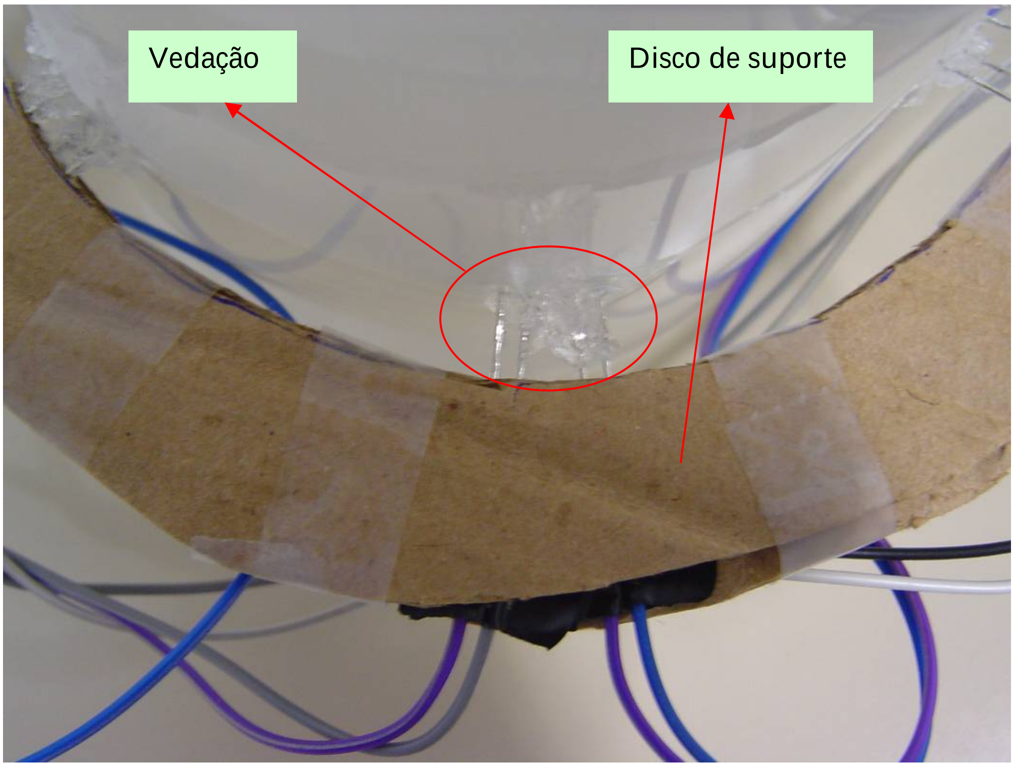



**Ilustração 6: Detalhe da vedação e do disco de suporte (matriz de emissores e receptores).**

Um cuidado importante a ser tomado durante a aquisição de dados é isolar o protótipo opticamente do ambiente. Isto é, deixá-lo livre da influência de luz externa (potencialmente infravermelha). Esse cuidado é facilmente tomado ao se tampar o recipiente e ao se colocar o mesmo dentro de uma caixa, como mostrado na Ilustração 7:

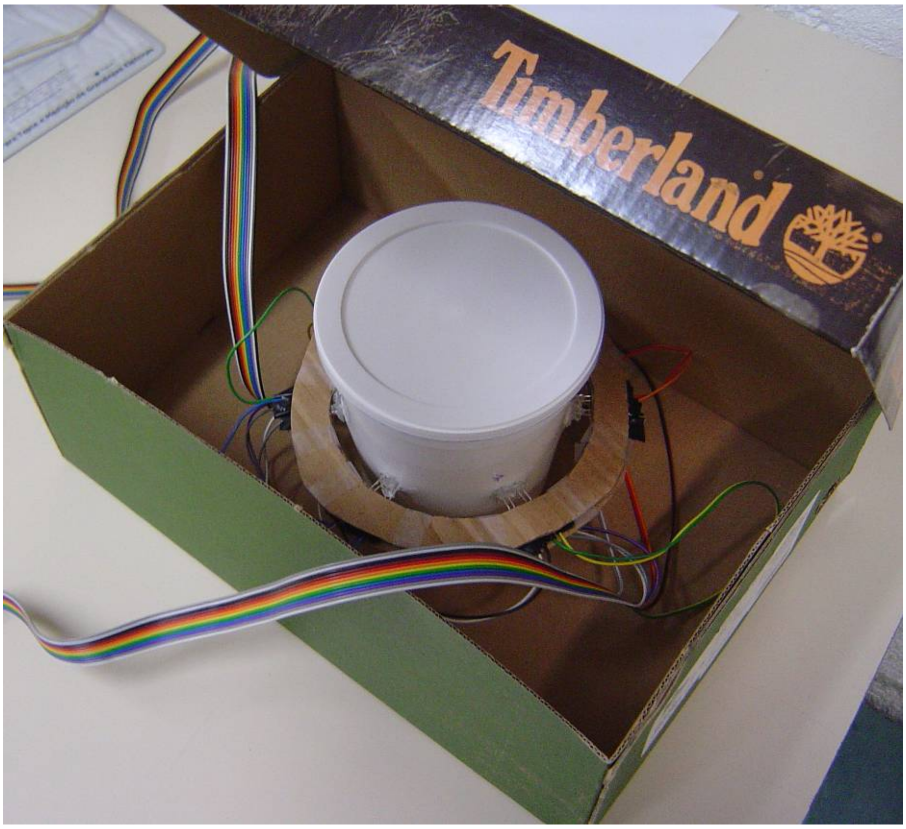



**Ilustração 7: Isolamento óptico do protótipo.**

Cada um dos doze componentes ópticos foi ligado através de *flat cables* até o circuito elétrico intermediário, o qual faz a interface com a placa de aquisição de dados. Esse circuito é detalhado na seção 3.2.

Finalmente, uma última providência foi tomada para garantir que o protótipo não tivesse sua funcionalidade comprometida: sua parte interna foi pintada de preto. O objetivo dessa medida é minimizar a reflexão dos raios infravermelhos nas paredes do recipiente, o que poderia perturbar as leituras. A tinta usada é à óleo, para garantir que não se deslocasse quando a cuba fosse preenchida com água. O resultado final pode ser visto na Ilustração 8:

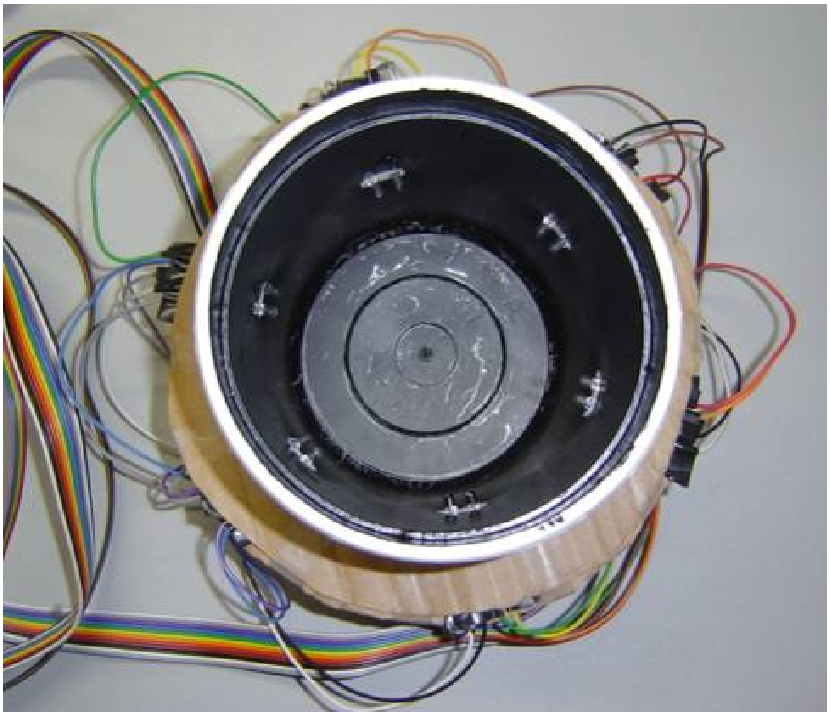


**Ilustração 8: Pintura final em preto, para evitar reflexão.**

## 3.2 Circuito Elétrico Intermediário

A configuração do circuito elétrico ao qual cada um dos LED's (e cada um dos fotodiodos) está ligado é igual. Portanto, aqui será apresentada apenas a ligação para um LED (e um fotodiodo).

A Ilustração 9 mostra a ligação entre um LED e a saída digital da placa de aquisição de dados:

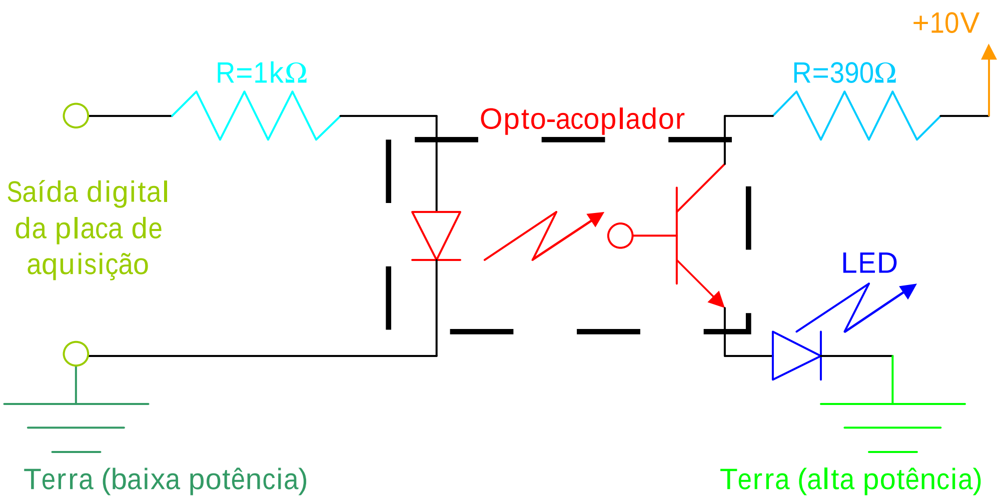



**Ilustração 9: Circuito elétrico dos emissores de luz.**

Como se pode verificar, o circuito da placa de aquisição está eletricamente isolado do circuito do LED através do opto-acoplador 4N33. O objetivo desse componente é evitar que sobrecargas na alimentação do LED afetem a placa de aquisição, que é bastante frágil. Dessa forma, o circuito de alta potência (ligado aos LED's e, como se verá, aos fotodiodos também) fica isolado eletricamente do circuito de baixa potência (ligado à placa de aquisição).

A tensão de +10V é obtida através de um regulador de tensão 7810. O resistor de 390Ω foi projetado para garantir uma corrente de 20mA no LED.

A saída digital da placa de aquisição segue o padrão TTL, daí o resistor de 1kΩ ligado em série a ela. Ele garante que a corrente não ultrapasse 5mA.

Foi tomado cuidado para que todos os fios ligando cada um dos 6 LED's tivessem o mesmo comprimento, reduzindo, assim, as diferenças entre resistências não consideradas no projeto. Isso foi facilmente conseguido com o uso de *flat cable*.

A ligação entre um fotodiodo e a entrada analógica da placa de aquisição de dados é mostrada na Ilustração 10:

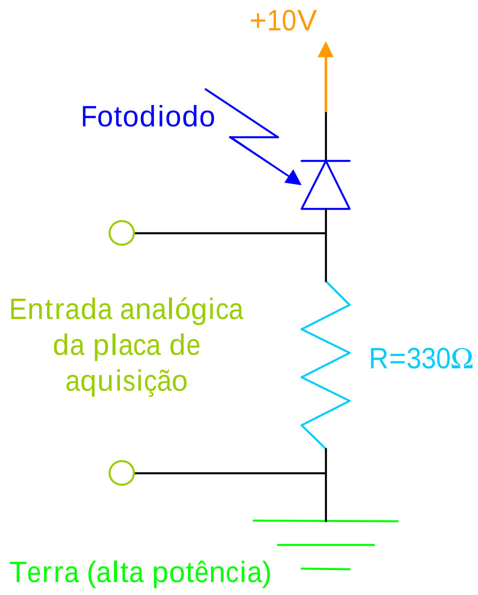



**Ilustração 10: Circuito elétrico dos receptores de luz.**

Como se pode ver, o fotodiodo está ligado com polaridade invertida. Logo, via de regra, ele vai suportar a tensão de +10V. Na escuridão total, a tensão sobre ele é realmente de +10V, de forma que não passa corrente no resistor. Quando incidem raios infravermelhos sobre ele, uma pequena corrente o atravessa, de forma que uma queda de tensão mensurável no resistor de 330Ω é detectada pela entrada analógica da placa de aquisição de dados. O resistor de 330Ω foi determinado experimentalmente, com base em medições preliminares, explicadas a seguir.

A Ilustração 11 mostra a montagem preliminar de um par emissor–receptor, usada para determinar o valor da resistência:

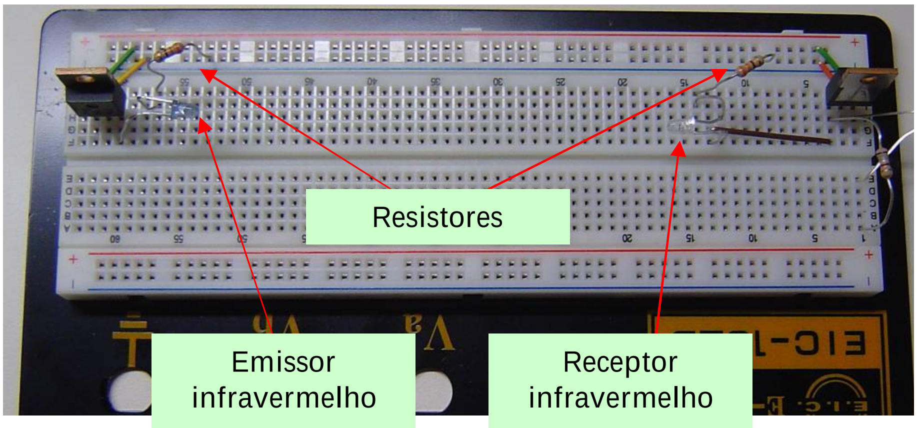



**Ilustração 11: Circuito de emissão e recepção de infravermelho.**

O emissor e o receptor estão virados um para o outro porque os raios infravermelhos são emitidos praticamente só na direção da cabeça do emissor. No ar, quase não ocorre espalhamento desses feixes. Portanto, essa configuração facilita a captação dos raios pelo receptor.

Para testar a capacidade de reação a estímulo do receptor, mediu-se a diferença de potencial no resistor que está em série com o mesmo. Vários valores de resistência foram testados aqui. Além disso, tanto o emissor como o receptor foram alimentados com vários reguladores de tensão entre +5V e +12V. E a experiência foi realizada no escuro, para evitar a interferência de fontes externas de raios infravermelhos. Após os testes, a configuração de tensão de +10V e resistência de 330Ω provou ser bastante interessante, apresentando as seguintes medidas:

- Escuridão total, com o emissor desligado. Mediu-se 0,1mV de queda de tensão no resistor.
- Claridade máxima, isto é, com o emissor ligado a 1cm do receptor. Mediu-se 150mV.
- Situação média, isto é, com o emissor ligado a 10cm do receptor (configuração mostrada na Ilustração 11). Mediu-se 50mV.
- Situação de trabalho, isto é, igual à configuração anterior, mas com um copo contendo uma mistura de água e leite entre emissor e receptor. Mediu-se 25mV.

Assim como no caso dos LED's, foi tomado cuidado para que todos os fios ligando cada um dos 6 fotodiodos tivessem o mesmo comprimento.

A execução do projeto descrito foi feita em uma única *protoboard*, conforme mostrado pela Ilustração 12. Na e na , podem-se ver detalhes de cada região da placa:

- Aquela responsável pela emissão dos sinais, e que está conectada, de um lado, à saída digital da placa de aquisição, e de outro, aos LED's.
- Aquela responsável pela coleta dos sinais, e que está conectada, de um lado, à entrada analógica da placa de aquisição, e de outro, aos fotodiodos.

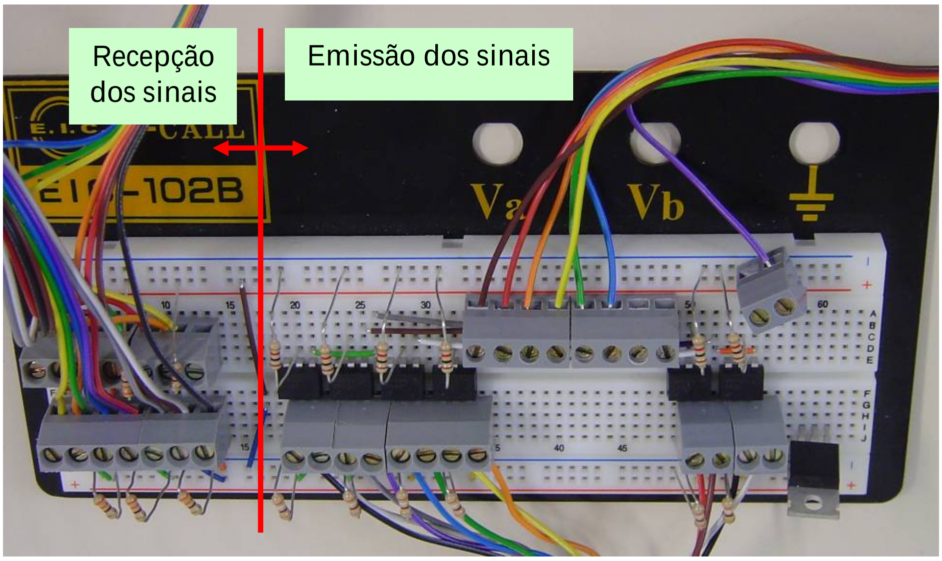



**Ilustração 12: Visão geral da *protoboard* que liga o protótipo à placa de aquisição.**

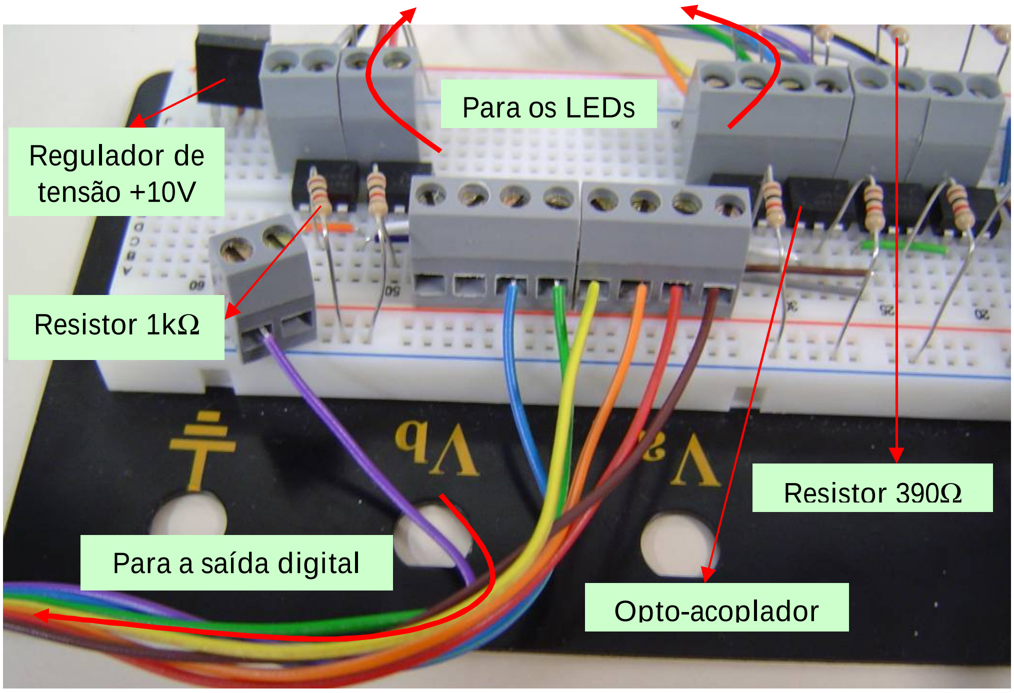



**Ilustração 13: Detalhe da região da *protoboard* responsável pela emissão de sinais.**

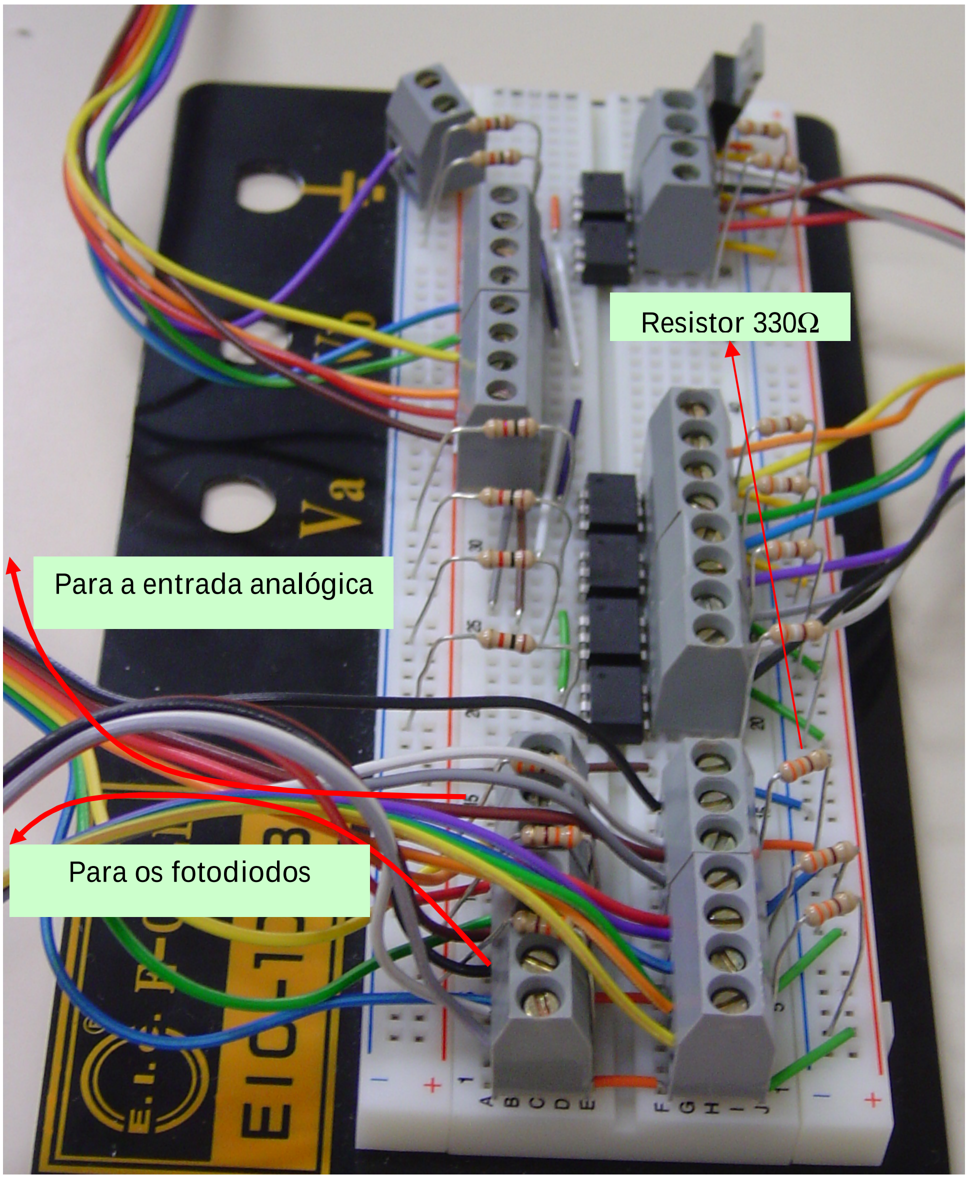



**Ilustração 14: Detalhe da região da *protoboard* responsável pela coleta de sinais.**

## 3.3 Placa de Aquisição de Dados

A placa de aquisição utilizada é do tipo PCL711, fabricada pela Advantech Co. Ltd. Dentre as suas características relevantes ao projeto, menciona-se:

- 8 canais de entrada analógica. Cada um deles pode medir valores entre ±0,3125V com 12 bits de precisão.
- 12 canais de saída digital que seguem o padrão TTL (fornecendo, portanto, até +5V).

A placa é conectada ao computador através de um *bus* do tipo ISA, o que tornou necessário o uso de um computador com placa-mãe que o suporte. O computador em questão é um Pentium PRO de 200MHz, rodando Linux Fedora 6. Embora ele não seja rápido, é suficiente para realizar o controle da placa e a aquisição de dados. As ligações com a *protoboard* são mostradas na Ilustração 15:

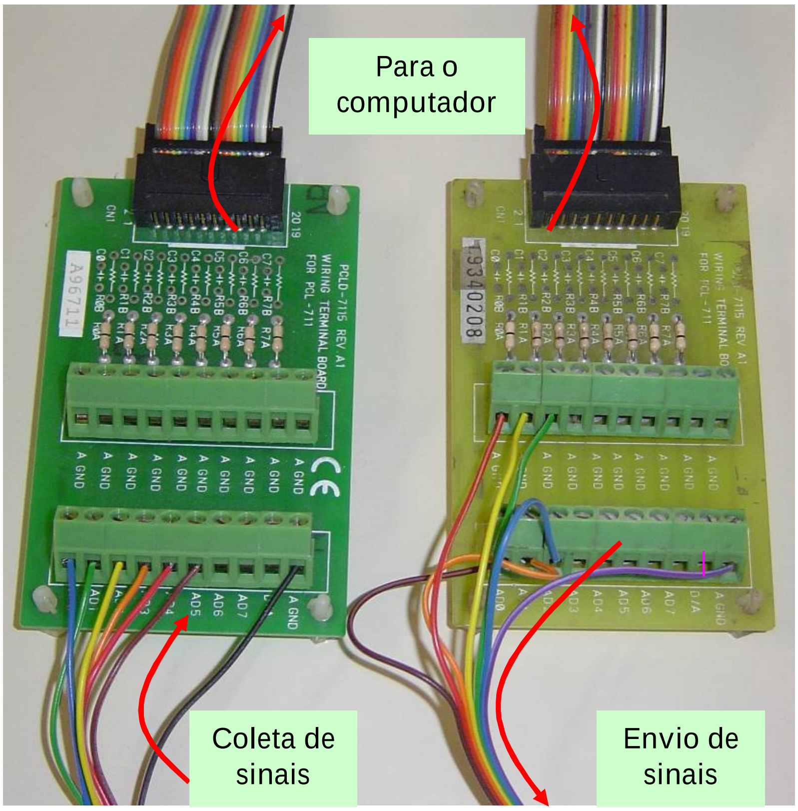



**Ilustração 15: Ligações da placa de aquisição com a *protoboard*.**

## 3.4 ***Software* de Controle da Placa**

A placa de aquisição pode ser controlada diretamente através de um programa escrito pelo usuário. Os canais de leitura e de saída de dados são acessíveis através de portas pré-definidas.

O objetivo do programa é realizar a seguinte série de instruções:

- Com os LED's todos desligados, ler cada um dos 6 canais de aquisição.
- Ligando um e apenas um LED, repetir essa leitura.
- Repetir a operação anterior para cada um dos seis LED's.
- Registrar todas as leituras em um arquivo de texto separado.

O código-fonte foi escrito em C, compilado, e executado com sucesso, conforme será discutido nos capítulos posteriores. O texto do código-fonte encontra-se no ANEXO A.

# 4. *HARDWARE*: RESULTADOS EXPERIMENTAIS (MEDIDAS)

Como mencionado no capítulo anterior, o programa de controle da placa de aquisição foi executado, gerando assim um documento com as leituras dos sinais.

## 4.1 Apresentação das medidas realizadas

Cada uma das situações foi amostrada 20 vezes. Segue o resultado de cada uma delas. Todos os valores estão em volts.

**Tabela 1: Dados adquiridos com os LED's desligados.**

| LEDs desligados | | | | | | |
|---|---|---|---|---|---|---|
| Medida | Canal 1 | Canal 2 | Canal 3 | Canal 4 | Canal 5 | Canal 6 |
| 0 | -0,001 | 0,000 | -0,001 | -0,001 | 0,000 | 0,000 |
| 1 | 0,000 | -0,001 | -0,001 | 0,000 | 0,000 | 0,000 |
| 2 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 |
| 3 | -0,001 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 |
| 4 | 0,000 | -0,001 | 0,000 | 0,000 | 0,000 | 0,000 |
| 5 | -0,001 | -0,001 | -0,001 | 0,000 | 0,000 | 0,000 |
| 6 | 0,000 | -0,001 | -0,001 | -0,001 | -0,001 | 0,000 |
| 7 | 0,000 | 0,000 | 0,000 | -0,001 | -0,001 | -0,001 |
| 8 | -0,001 | 0,000 | 0,000 | -0,001 | -0,001 | 0,000 |
| 9 | -0,001 | 0,000 | 0,000 | 0,000 | 0,000 | -0,001 |
| 10 | 0,000 | 0,000 | -0,001 | 0,000 | 0,000 | -0,001 |
| 11 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 |
| 12 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 |
| 13 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 |
| 14 | 0,000 | -0,001 | -0,001 | -0,001 | -0,001 | 0,000 |
| 15 | 0,000 | -0,001 | -0,001 | -0,001 | 0,000 | 0,000 |
| 16 | 0,000 | 0,000 | 0,000 | 0,000 | -0,001 | 0,000 |
| 17 | 0,000 | -0,001 | 0,000 | 0,000 | 0,000 | -0,001 |
| 18 | 0,000 | 0,000 | 0,000 | -0,001 | 0,000 | 0,000 |
| 19 | -0,001 | -0,001 | 0,000 | 0,000 | 0,000 | 0,000 |
| **Média** | **0,000** | **0,000** | **0,000** | **0,000** | **0,000** | **0,000** |
| **Desvio** | **0,000** | **0,000** | **0,000** | **0,000** | **0,000** | **0,000** |

Tabela 2: Dados adquiridos com apenas o LED 1 ativado.

| LED 1 ativado | | | | | | |
|---|---|---|---|---|---|---|
| Medida | Canal 1 | Canal 2 | Canal 3 | Canal 4 | Canal 5 | Canal 6 |
| 0 | 0,001 | 0,008 | 0,006 | 0,012 | 0,008 | 0,010 |
| 1 | 0,136 | 0,008 | 0,006 | 0,012 | 0,008 | 0,010 |
| 2 | 0,137 | 0,008 | 0,006 | 0,012 | 0,008 | 0,010 |
| 3 | 0,136 | 0,009 | 0,006 | 0,012 | 0,008 | 0,010 |
| 4 | 0,136 | 0,009 | 0,005 | 0,012 | 0,008 | 0,010 |
| 5 | 0,137 | 0,007 | 0,006 | 0,013 | 0,008 | 0,010 |
| 6 | 0,137 | 0,008 | 0,006 | 0,012 | 0,009 | 0,010 |
| 7 | 0,137 | 0,008 | 0,006 | 0,013 | 0,008 | 0,010 |
| 8 | 0,137 | 0,007 | 0,006 | 0,012 | 0,008 | 0,010 |
| 9 | 0,137 | 0,008 | 0,006 | 0,013 | 0,009 | 0,011 |
| 10 | 0,137 | 0,009 | 0,006 | 0,012 | 0,009 | 0,011 |
| 11 | 0,137 | 0,007 | 0,005 | 0,012 | 0,009 | 0,011 |
| 12 | 0,136 | 0,008 | 0,006 | 0,013 | 0,009 | 0,011 |
| 13 | 0,136 | 0,007 | 0,005 | 0,013 | 0,008 | 0,010 |
| 14 | 0,137 | 0,008 | 0,006 | 0,012 | 0,008 | 0,010 |
| 15 | 0,137 | 0,007 | 0,006 | 0,013 | 0,008 | 0,010 |
| 16 | 0,136 | 0,008 | 0,005 | 0,012 | 0,008 | 0,010 |
| 17 | 0,137 | 0,008 | 0,006 | 0,012 | 0,008 | 0,010 |
| 18 | 0,137 | 0,008 | 0,006 | 0,012 | 0,008 | 0,011 |
| 19 | 0,137 | 0,008 | 0,006 | 0,012 | 0,008 | 0,010 |
| **Média** | **0,130** | **0,008** | **0,006** | **0,012** | **0,008** | **0,010** |
| **Desvio** | **0,030** | **0,000** | **0,000** | **0,000** | **0,000** | **0,000** |

Tabela 3: Dados adquiridos com apenas o LED 2 ativado.

| LED 2 ativado | | | | | | |
|---|---|---|---|---|---|---|
| Medida | Canal 1 | Canal 2 | Canal 3 | Canal 4 | Canal 5 | Canal 6 |
| 0 | 0,146 | 0,113 | 0,043 | 0,006 | 0,018 | 0,025 |
| 1 | 0,070 | 0,114 | 0,043 | 0,007 | 0,018 | 0,025 |
| 2 | 0,030 | 0,113 | 0,043 | 0,007 | 0,018 | 0,025 |
| 3 | 0,016 | 0,114 | 0,044 | 0,007 | 0,018 | 0,025 |
| 4 | 0,012 | 0,113 | 0,044 | 0,007 | 0,018 | 0,025 |
| 5 | 0,011 | 0,113 | 0,043 | 0,006 | 0,018 | 0,025 |
| 6 | 0,010 | 0,113 | 0,043 | 0,007 | 0,018 | 0,026 |
| 7 | 0,011 | 0,114 | 0,043 | 0,007 | 0,018 | 0,025 |
| 8 | 0,011 | 0,114 | 0,043 | 0,007 | 0,018 | 0,025 |
| 9 | 0,012 | 0,114 | 0,043 | 0,007 | 0,018 | 0,025 |
| 10 | 0,011 | 0,114 | 0,043 | 0,007 | 0,018 | 0,024 |
| 11 | 0,010 | 0,113 | 0,043 | 0,007 | 0,018 | 0,025 |
| 12 | 0,011 | 0,114 | 0,043 | 0,007 | 0,018 | 0,025 |
| 13 | 0,011 | 0,114 | 0,043 | 0,007 | 0,018 | 0,025 |
| 14 | 0,011 | 0,113 | 0,043 | 0,007 | 0,017 | 0,025 |
| 15 | 0,011 | 0,114 | 0,043 | 0,007 | 0,017 | 0,026 |
| 16 | 0,010 | 0,113 | 0,043 | 0,007 | 0,018 | 0,025 |
| 17 | 0,011 | 0,114 | 0,043 | 0,006 | 0,018 | 0,025 |
| 18 | 0,011 | 0,113 | 0,043 | 0,006 | 0,018 | 0,025 |
| 19 | 0,011 | 0,113 | 0,042 | 0,007 | 0,018 | 0,025 |
| **Média** | **0,022** | **0,113** | **0,043** | **0,007** | **0,018** | **0,025** |
| **Desvio** | **0,031** | **0,000** | **0,000** | **0,000** | **0,000** | **0,000** |

Tabela 4: Dados adquiridos com apenas o LED 3 ativado.

| LED 3 ativado | | | | | | |
|---|---|---|---|---|---|---|
| Medida | Canal 1 | Canal 2 | Canal 3 | Canal 4 | Canal 5 | Canal 6 |
| 0 | 0,024 | 0,009 | 0,143 | 0,009 | 0,008 | 0,033 |
| 1 | 0,024 | 0,008 | 0,144 | 0,008 | 0,008 | 0,034 |
| 2 | 0,020 | 0,009 | 0,144 | 0,009 | 0,008 | 0,034 |
| 3 | 0,019 | 0,009 | 0,144 | 0,009 | 0,008 | 0,033 |
| 4 | 0,020 | 0,008 | 0,144 | 0,008 | 0,009 | 0,034 |
| 5 | 0,019 | 0,008 | 0,145 | 0,008 | 0,009 | 0,034 |
| 6 | 0,020 | 0,008 | 0,144 | 0,008 | 0,008 | 0,034 |
| 7 | 0,019 | 0,008 | 0,145 | 0,009 | 0,008 | 0,034 |
| 8 | 0,019 | 0,008 | 0,144 | 0,009 | 0,008 | 0,034 |
| 9 | 0,019 | 0,008 | 0,145 | 0,008 | 0,008 | 0,033 |
| 10 | 0,019 | 0,008 | 0,145 | 0,008 | 0,008 | 0,034 |
| 11 | 0,018 | 0,008 | 0,143 | 0,008 | 0,008 | 0,033 |
| 12 | 0,020 | 0,009 | 0,144 | 0,008 | 0,008 | 0,033 |
| 13 | 0,020 | 0,009 | 0,144 | 0,009 | 0,008 | 0,034 |
| 14 | 0,019 | 0,009 | 0,144 | 0,009 | 0,008 | 0,033 |
| 15 | 0,020 | 0,009 | 0,144 | 0,009 | 0,009 | 0,033 |
| 16 | 0,020 | 0,009 | 0,144 | 0,008 | 0,008 | 0,034 |
| 17 | 0,018 | 0,008 | 0,144 | 0,009 | 0,008 | 0,033 |
| 18 | 0,019 | 0,008 | 0,144 | 0,008 | 0,008 | 0,034 |
| 19 | 0,019 | 0,008 | 0,145 | 0,009 | 0,008 | 0,033 |
| **Média** | **0,020** | **0,008** | **0,144** | **0,008** | **0,008** | **0,034** |
| **Desvio** | **0,002** | **0,000** | **0,000** | **0,000** | **0,000** | **0,000** |

Tabela 5: Dados adquiridos com apenas o LED 4 ativado.

| LED 4 ativado | | | | | | |
|---|---|---|---|---|---|---|
| Medida | Canal 1 | Canal 2 | Canal 3 | Canal 4 | Canal 5 | Canal 6 |
| 0 | 0,034 | 0,012 | 0,007 | 0,093 | 0,007 | 0,019 |
| 1 | 0,027 | 0,013 | 0,007 | 0,093 | 0,007 | 0,019 |
| 2 | 0,021 | 0,012 | 0,007 | 0,093 | 0,007 | 0,019 |
| 3 | 0,019 | 0,012 | 0,007 | 0,093 | 0,007 | 0,020 |
| 4 | 0,019 | 0,013 | 0,007 | 0,092 | 0,007 | 0,019 |
| 5 | 0,018 | 0,012 | 0,007 | 0,093 | 0,007 | 0,020 |
| 6 | 0,019 | 0,012 | 0,007 | 0,093 | 0,006 | 0,019 |
| 7 | 0,019 | 0,012 | 0,007 | 0,093 | 0,007 | 0,020 |
| 8 | 0,019 | 0,012 | 0,007 | 0,093 | 0,007 | 0,019 |
| 9 | 0,019 | 0,013 | 0,007 | 0,093 | 0,008 | 0,019 |
| 10 | 0,019 | 0,012 | 0,007 | 0,091 | 0,007 | 0,019 |
| 11 | 0,019 | 0,013 | 0,007 | 0,093 | 0,007 | 0,020 |
| 12 | 0,019 | 0,013 | 0,007 | 0,093 | 0,007 | 0,019 |
| 13 | 0,019 | 0,012 | 0,007 | 0,093 | 0,007 | 0,019 |
| 14 | 0,020 | 0,013 | 0,007 | 0,093 | 0,007 | 0,019 |
| 15 | 0,019 | 0,012 | 0,007 | 0,093 | 0,007 | 0,019 |
| 16 | 0,020 | 0,013 | 0,007 | 0,093 | 0,007 | 0,019 |
| 17 | 0,019 | 0,013 | 0,007 | 0,093 | 0,007 | 0,019 |
| 18 | 0,019 | 0,012 | 0,007 | 0,093 | 0,007 | 0,019 |
| 19 | 0,019 | 0,013 | 0,007 | 0,093 | 0,007 | 0,019 |
| **Média** | **0,020** | **0,012** | **0,007** | **0,093** | **0,007** | **0,019** |
| **Desvio** | **0,004** | **0,000** | **0,000** | **0,000** | **0,000** | **0,000** |

**Tabela 6: Dados adquiridos com apenas o LED 5 ativado.**

| LED 5 ativado | | | | | | |
|---|---|---|---|---|---|---|
| Medida | Canal 1 | Canal 2 | Canal 3 | Canal 4 | Canal 5 | Canal 6 |
| 0 | 0,026 | 0,014 | 0,008 | 0,005 | 0,108 | 0,014 |
| 1 | 0,017 | 0,015 | 0,008 | 0,005 | 0,109 | 0,014 |
| 2 | 0,010 | 0,014 | 0,008 | 0,005 | 0,109 | 0,014 |
| 3 | 0,008 | 0,015 | 0,008 | 0,005 | 0,109 | 0,014 |
| 4 | 0,007 | 0,015 | 0,008 | 0,005 | 0,109 | 0,014 |
| 5 | 0,007 | 0,015 | 0,009 | 0,005 | 0,109 | 0,014 |
| 6 | 0,007 | 0,015 | 0,008 | 0,005 | 0,108 | 0,014 |
| 7 | 0,007 | 0,015 | 0,009 | 0,004 | 0,109 | 0,014 |
| 8 | 0,007 | 0,015 | 0,008 | 0,004 | 0,109 | 0,014 |
| 9 | 0,007 | 0,015 | 0,008 | 0,005 | 0,109 | 0,014 |
| 10 | 0,007 | 0,014 | 0,009 | 0,005 | 0,108 | 0,014 |
| 11 | 0,006 | 0,015 | 0,008 | 0,004 | 0,109 | 0,014 |
| 12 | 0,007 | 0,015 | 0,008 | 0,004 | 0,108 | 0,014 |
| 13 | 0,007 | 0,015 | 0,009 | 0,005 | 0,108 | 0,013 |
| 14 | 0,006 | 0,015 | 0,009 | 0,005 | 0,108 | 0,014 |
| 15 | 0,007 | 0,015 | 0,008 | 0,004 | 0,108 | 0,014 |
| 16 | 0,007 | 0,015 | 0,008 | 0,004 | 0,109 | 0,015 |
| 17 | 0,007 | 0,014 | 0,008 | 0,005 | 0,109 | 0,014 |
| 18 | 0,007 | 0,014 | 0,008 | 0,005 | 0,108 | 0,014 |
| 19 | 0,007 | 0,014 | 0,008 | 0,005 | 0,109 | 0,014 |
| **Média** | **0,009** | **0,015** | **0,008** | **0,005** | **0,109** | **0,014** |
| **Desvio** | **0,005** | **0,000** | **0,000** | **0,000** | **0,000** | **0,000** |

**Tabela 7: Dados adquiridos com apenas o LED 6 ativado.**

| LED 6 ativado | | | | | | |
|---|---|---|---|---|---|---|
| Medida | Canal 1 | Canal 2 | Canal 3 | Canal 4 | Canal 5 | Canal 6 |
| 0 | 0,019 | 0,007 | 0,011 | 0,006 | 0,004 | 0,171 |
| 1 | 0,038 | 0,006 | 0,011 | 0,006 | 0,004 | 0,173 |
| 2 | 0,036 | 0,007 | 0,012 | 0,007 | 0,005 | 0,174 |
| 3 | 0,035 | 0,007 | 0,012 | 0,007 | 0,004 | 0,173 |
| 4 | 0,033 | 0,006 | 0,012 | 0,006 | 0,005 | 0,173 |
| 5 | 0,034 | 0,007 | 0,012 | 0,007 | 0,005 | 0,174 |
| 6 | 0,034 | 0,007 | 0,011 | 0,007 | 0,004 | 0,173 |
| 7 | 0,034 | 0,006 | 0,012 | 0,007 | 0,005 | 0,175 |
| 8 | 0,035 | 0,006 | 0,012 | 0,007 | 0,005 | 0,174 |
| 9 | 0,035 | 0,006 | 0,012 | 0,007 | 0,004 | 0,173 |
| 10 | 0,034 | 0,007 | 0,011 | 0,006 | 0,005 | 0,174 |
| 11 | 0,034 | 0,007 | 0,011 | 0,007 | 0,005 | 0,174 |
| 12 | 0,034 | 0,007 | 0,012 | 0,007 | 0,005 | 0,174 |
| 13 | 0,035 | 0,007 | 0,012 | 0,007 | 0,005 | 0,175 |
| 14 | 0,035 | 0,007 | 0,012 | 0,007 | 0,004 | 0,175 |
| 15 | 0,034 | 0,007 | 0,012 | 0,007 | 0,005 | 0,174 |
| 16 | 0,034 | 0,007 | 0,012 | 0,007 | 0,004 | 0,174 |
| 17 | 0,034 | 0,007 | 0,012 | 0,006 | 0,004 | 0,175 |
| 18 | 0,034 | 0,007 | 0,012 | 0,006 | 0,005 | 0,175 |
| 19 | 0,034 | 0,007 | 0,012 | 0,007 | 0,005 | 0,175 |
| **Média** | **0,034** | **0,007** | **0,012** | **0,007** | **0,005** | **0,174** |
| **Desvio** | **0,004** | **0,000** | **0,000** | **0,000** | **0,000** | **0,001** |

Os dados acima podem ser compilados em uma única tabela de médias dos valores. Como se pôde ver, os desvios-padrão são sempre ínfimos, o que confere significado estatístico a esse resumo, que se encontra na Tabela 8. Vale mencionar aqui que a numeração dos LED’s e dos canais segue o seguinte padrão: o LED de número “n” encontra-se diametralmente oposto ao fotodiodo ligado ao canal “n”.

**Tabela 8: Resumo dos dados coletados.**

| Média | Canal 1 | Canal 2 | Canal 3 | Canal 4 | Canal 5 | Canal 6 |
|---|---|---|---|---|---|---|
| LED 1 | 0,130 | 0,008 | 0,006 | 0,012 | 0,008 | 0,010 |
| LED 2 | 0,022 | 0,113 | 0,043 | 0,007 | 0,018 | 0,025 |
| LED 3 | 0,020 | 0,008 | 0,144 | 0,008 | 0,008 | 0,034 |
| LED 4 | 0,020 | 0,012 | 0,007 | 0,093 | 0,007 | 0,019 |
| LED 5 | 0,009 | 0,015 | 0,008 | 0,005 | 0,109 | 0,014 |
| LED 6 | 0,034 | 0,007 | 0,012 | 0,007 | 0,005 | 0,174 |

## 4.2 Discussão acerca das medidas

Na Tabela 8, ficou clara a correlação entre as emissões de infravermelho de cada um dos LED’s e a leitura mensurável nos canais correspondentes. De fato, quando um LED é aceso, lêem-se valores da ordem de 0,1V no receptor diametralmente oposto, ao passo que os outros fotodiodos registram em torno de 0,01V. Essa diferença de 10 vezes prova que o *hardware* é capaz de informar confiavelmente as condições de iluminação em cada um dos receptores de luz.

Esse fato comprova a aplicabilidade do protótipo para o fim proposto nos objetivos deste Trabalho. O próximo passo a ser tomado para finalização da construção do *hardware* consiste na calibração de cada um dos canais. Como se pode observar na tabela mencionada, dentro de cada uma das colunas, os valores são coerentes. Isto é, quando o fotodiodo é diretamente iluminado, ele claramente fornece um valor de tensão bem diferentes dos outros valores da mesma coluna. Entretanto, dentro de cada uma das linhas, os valores não são coerentes. Por exemplo, no canal 6 lê-se 0,174V quando ele é iluminado, mas no canal 4, apenas 0,093V. Essa diferença de 47% no valor se deve às diferenças de fabricação entre os fotodiodos e resistores utilizados (vale mencionar que, a princípio, eles eram todos iguais, mas diferenças aleatórias ocorrem no processo de fabricação).

Dessa forma, verifica-se que o aparato é funcional e é capaz de cumprir seu objetivo, mas precisa ser calibrado para fornecer medidas mais confiáveis.

# 5. *SOFTWARE*: MÉTODO DOS ELEMENTOS FINITOS

## 5.1 **Resolução do problema direto: princípio variacional**

Conforme explicado no capítulo 2, o fenômeno físico de difusão de fótons pode ser descrita pela equação de Helmholtz, que é aqui reproduzida:

$$-\nabla^2(U)+\eta U=\frac{q_0}{\sqrt{D}} \quad \textbf{(5.1)}$$

Para resolver essa equação, emprega-se um princípio variacional. Segundo [8], uma equação que pode ser escrita na forma $\underline{L}\underline{u}+\underline{b}=\underline{0}$, onde $\underline{L}$ é um operador auto-adjunto, admite um funcional do tipo $\Pi=\iiint_{\Omega}\left[\frac{1}{2}\underline{u}^T\underline{L}\underline{u}+\underline{u}^T\underline{b}\right]d\Omega$.

A equação de Helmholtz não-homogênea pode ser escrita na forma acima, fazendo-se $\underline{L}=\left[\frac{\partial^2}{\partial x^2};\frac{\partial^2}{\partial y^2};\frac{\partial^2}{\partial z^2};\eta\right]$ e $\underline{b}=\frac{q_0}{\sqrt{D}}$. Portanto, admite o seguinte funcional:

$$\Pi=\iiint_{\Omega}\left[\frac{1}{2}U\left(\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}+\frac{\partial^2 U}{\partial z^2}+\eta U\right)+\frac{q_0}{\sqrt{D}}U\right]d\Omega \quad \textbf{(5.2)}$$

Integrando por partes os três primeiros termos, chega-se a:

$$\Pi=-\iiint_{\Omega}\left[\frac{1}{2}\left(\left\{\frac{\partial U}{\partial x}\right\}^2+\left\{\frac{\partial U}{\partial y}\right\}^2+\left\{\frac{\partial U}{\partial z}\right\}^2-\eta U^2\right)-\frac{q_0}{\sqrt{D}}U\right]d\Omega \quad \textbf{(5.3)}$$

Deseja-se resolver o problema estacionário, isto é, sem variação temporal na intensidade de luz emitida e medida. Logo, o termo $\eta$ pode ser assim simplificado:

$$\eta=\eta(r)=\frac{\mu_a}{D}+\frac{\nabla^2\left(\sqrt{D}\right)}{\sqrt{D}}+\frac{1}{cD}\frac{\partial}{\partial t}=\frac{\mu_a}{D}+\frac{\nabla^2\left(\sqrt{D}\right)}{\sqrt{D}} \quad \textbf{(5.4)}$$

Além disso, o coeficiente de difusão $D$ é constante dentro de cada elemento finito. Há duas conseqüências para esse fato. Em primeiro lugar, seu Laplaciano é nulo:

$$\eta = \eta(r) = \frac{\mu_a}{D} + \frac{\nabla^2\left(\sqrt{D}\right)}{\sqrt{D}} = \frac{\mu_a}{D} \qquad \textbf{(5.5)}$$

Em segundo lugar, a função $U$ pode ser desmembrada em $\sqrt{D}\Phi$. Aplicando esse fato na expressão do funcional:

$$\Pi = -\iiint_{\Omega}\left[\frac{1}{2}\left(\left\{\frac{\partial\sqrt{D}\Phi}{\partial x}\right\}^2 + \left\{\frac{\partial\sqrt{D}\Phi}{\partial y}\right\}^2 + \left\{\frac{\partial\sqrt{D}\Phi}{\partial z}\right\}^2 - \frac{\mu_a}{D}U^2\right) - \frac{q_0}{\sqrt{D}}U\right]d\Omega \qquad \textbf{(5.6)}$$

$$\Pi = -\iiint_{\Omega}\left[\frac{D}{2}\left(\left\{\frac{\partial\Phi}{\partial x}\right\}^2 + \left\{\frac{\partial\Phi}{\partial y}\right\}^2 + \left\{\frac{\partial\Phi}{\partial z}\right\}^2\right) - \frac{\mu_a}{2}\Phi^2 - q_0\Phi\right]d\Omega \qquad \textbf{(5.7)}$$

Com esse funcional em mãos, é possível determinar $\Phi$: ele será a função que anula a derivada (em relação ao próprio $\Phi$) de $\Pi$.

## 5.2 **Resolução do problema direto: discretização do domínio**

A formulação matemática da discretização da função $\Phi$ será do tipo bilinear (pois se consideram grandezas físicas constantes na direção $z$) e os elementos finitos serão triangulares, conforme recomenda [9]. Isto é, a função $\Phi$ será aproximada por:

$$\Phi \approx \hat{\Phi} = a_1 + a_2 x + a_3 y \qquad \textbf{(5.8)}$$

Deseja-se que $\hat{\Phi}$ seja idêntica a $\Phi$ pelo menos nos nós $(i, j, m)$ do elemento finito. Como este é triangular, essa condição equivale a:

$$\begin{cases} \Phi(x_i, y_i) = \phi_i = \hat{\Phi}(x_i, y_i) = a_1 + a_2 x_i + a_3 y_i \\ \Phi(x_j, y_j) = \phi_j = \hat{\Phi}(x_j, y_j) = a_1 + a_2 x_j + a_3 y_j \\ \Phi(x_m, y_m) = \phi_m = \hat{\Phi}(x_m, y_m) = a_1 + a_2 x_m + a_3 y_m \end{cases} \tag{5.9}$$

$$\therefore \begin{bmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \phi_i \\ \phi_j \\ \phi_m \end{bmatrix} \tag{5.10}$$

$$\therefore \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{bmatrix}^{-1} \begin{bmatrix} \phi_i \\ \phi_j \\ \phi_m \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} \alpha_i & \alpha_j & \alpha_m \\ \beta_i & \beta_j & \beta_m \\ \gamma_i & \gamma_j & \gamma_m \end{bmatrix} \begin{bmatrix} \phi_i \\ \phi_j \\ \phi_m \end{bmatrix} \tag{5.11}$$

Aplicando a regra de Cramer, determina-se o valor de $2A$, $\alpha_i$, $\alpha_j$, $\alpha_m$, $\beta_i$, etc.:

$$2A = \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix} \tag{5.12}$$

Vale mencionar que $2A$ é a área do triângulo. Quanto às outras incógnitas:

$$\alpha_i = +\begin{vmatrix} x_j & y_j \\ x_m & y_m \end{vmatrix}; \beta_i = -\begin{vmatrix} 1 & y_j \\ 1 & y_m \end{vmatrix}; \gamma_i = +\begin{vmatrix} 1 & x_j \\ 1 & x_m \end{vmatrix} \tag{5.13}$$

$$\alpha_j = -\begin{vmatrix} x_i & y_i \\ x_m & y_m \end{vmatrix}; \beta_j = +\begin{vmatrix} 1 & y_i \\ 1 & y_m \end{vmatrix}; \gamma_j = -\begin{vmatrix} 1 & x_i \\ 1 & x_m \end{vmatrix} \tag{5.14}$$

$$\alpha_m = +\begin{vmatrix} x_i & y_i \\ x_j & y_j \end{vmatrix}; \beta_m = -\begin{vmatrix} 1 & y_i \\ 1 & y_j \end{vmatrix}; \gamma_m = +\begin{vmatrix} 1 & x_i \\ 1 & x_j \end{vmatrix} \tag{5.15}$$

Dessa forma, $\hat{\Phi}$ pode ser reescrita assim:

$$\Phi \approx \hat{\Phi} = a_1 + a_2 x + a_3 y = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & x & y \end{bmatrix} \frac{1}{2A} \begin{bmatrix} \alpha_i & \alpha_j & \alpha_m \\ \beta_i & \beta_j & \beta_m \\ \gamma_i & \gamma_j & \gamma_m \end{bmatrix} \begin{bmatrix} \phi_i \\ \phi_j \\ \phi_m \end{bmatrix} \tag{5.16}$$

Com essa expressão, a matriz linha das funções de forma fica definida:

$$\Phi \approx \hat{\Phi} = \begin{bmatrix} 1 & x & y \end{bmatrix} \frac{1}{2A} \begin{bmatrix} \alpha_i & \alpha_j & \alpha_m \\ \beta_i & \beta_j & \beta_m \\ \gamma_i & \gamma_j & \gamma_m \end{bmatrix} \begin{bmatrix} \phi_i \\ \phi_j \\ \phi_m \end{bmatrix} = \begin{bmatrix} N_i & N_j & N_m \end{bmatrix} \begin{bmatrix} \phi_i \\ \phi_j \\ \phi_m \end{bmatrix} = \underline{N}^T \underline{\phi} \quad \textbf{(5.17)}$$

Nessa expressão, $N_k = \frac{1}{2A}(\alpha_k + \beta_k x + \gamma_k y), \forall k \in \{i, j, m\}$.

Como as derivadas de $\Phi$ aparecem na expressão do funcional $\Pi$, o cálculo das mesmas será feito:

$$\begin{bmatrix} \frac{\partial \Phi}{\partial x} \\ \frac{\partial \Phi}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \Phi \approx \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \hat{\Phi} = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} N_i & N_j & N_m \end{bmatrix} \begin{bmatrix} \phi_i \\ \phi_j \\ \phi_m \end{bmatrix} \quad \textbf{(5.18)}$$

$$\therefore \begin{bmatrix} \frac{\partial \Phi}{\partial x} \\ \frac{\partial \Phi}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & \frac{\partial N_j}{\partial x} & \frac{\partial N_m}{\partial x} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_j}{\partial y} & \frac{\partial N_m}{\partial y} \end{bmatrix} \begin{bmatrix} \phi_i \\ \phi_j \\ \phi_m \end{bmatrix} \quad \textbf{(5.19)}$$

Pela definição dos $N_k, \forall k \in \{i, j, m\}$, calcula-se o valor de cada derivada acima, e define-se a matriz $\underline{\underline{B}}$:

$$\therefore \begin{bmatrix} \frac{\partial \Phi}{\partial x} \\ \frac{\partial \Phi}{\partial y} \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} \beta_i & \beta_j & \beta_m \\ \gamma_i & \gamma_j & \gamma_m \end{bmatrix} \begin{bmatrix} \phi_i \\ \phi_j \\ \phi_m \end{bmatrix} = \underline{\underline{B}} \underline{\phi} \quad \textbf{(5.20)}$$

## 5.3 **Resolução do problema direto: minimização do funcional**

Finalmente, pode-se aplicar a discretização do domínio na expressão do funcional para, em seguida, minimizá-lo e obter a função $\Phi$. Como o problema está sendo tratado como bidimensional, omite-se a derivada em $z$:

$$\Pi = -\iiint_{\Omega} \left[ \frac{D}{2} \left( \left\{ \frac{\partial \Phi}{\partial x} \right\}^2 + \left\{ \frac{\partial \Phi}{\partial y} \right\}^2 \right) - \frac{\mu_a}{2} \Phi^2 - q_0 \Phi \right] d\Omega \quad \textbf{(5.21)}$$

$$\Pi = -\iiint_{\Omega} \left[ \frac{D}{2} \begin{bmatrix} \frac{\partial \Phi}{\partial x} & \frac{\partial \Phi}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial \Phi}{\partial x} \\ \frac{\partial \Phi}{\partial y} \end{bmatrix} - \frac{\mu_a}{2} \Phi^T \Phi - q_0 \Phi \right] d\Omega \tag{5.22}$$

$$\Pi = -\iiint_{\Omega} \left[ \frac{D}{2} \underline{\phi}^T \underline{\underline{B}}^T \underline{\underline{I}}\, \underline{\underline{B}} \underline{\phi} - \frac{\mu_a}{2} \underline{\phi}^T \underline{N}\, \underline{N}^T \underline{\phi} - q_0 \underline{N}^T \underline{\phi} \right] d\Omega \tag{5.23}$$

$$\frac{\partial \Pi}{\partial \underline{\phi}} = -\iiint_{\Omega} \left[ \frac{D}{2} \underline{\phi}^T \left( \underline{\underline{B}}^T \underline{\underline{I}}\, \underline{\underline{B}} + \left( \underline{\underline{B}}^T \underline{\underline{I}}\, \underline{\underline{B}} \right)^T \right) - \frac{\mu_a}{2} \underline{\phi}^T \left( \underline{N}\, \underline{N}^T + \left( \underline{N}\, \underline{N}^T \right)^T \right) - q_0 \underline{N}^T \right] d\Omega \tag{5.24}$$

Devido à simetria das matrizes $\underline{\underline{B}}^T \underline{\underline{I}}\, \underline{\underline{B}}$ e $\underline{N}\, \underline{N}^T$ :

$$\frac{\partial \Pi}{\partial \underline{\phi}} = -\iiint_{\Omega} \left[ D \underline{\phi}^T \underline{\underline{B}}^T \underline{\underline{I}}\, \underline{\underline{B}} - \mu_a \underline{\phi}^T \underline{N}\, \underline{N}^T - q_0 \underline{N}^T \right] d\Omega \tag{5.25}$$

Relembrando que o objetivo aqui é minimizar o funcional:

$$\underline{0} = \frac{\partial \Pi}{\partial \underline{\phi}} = -\iiint_{\Omega} \left[ D \underline{\phi}^T \underline{\underline{B}}^T \underline{\underline{I}}\, \underline{\underline{B}} - \mu_a \underline{\phi}^T \underline{N}\, \underline{N}^T - q_0 \underline{N}^T \right] d\Omega \tag{5.26}$$

$$\underline{0}^T = -\iiint_{\Omega} \left[ D \underline{\underline{B}}^T \underline{\underline{I}}\, \underline{\underline{B}} - \mu_a \underline{N}\, \underline{N}^T \right] d\Omega \underline{\phi} + \iiint_{\Omega} q_0 \underline{N} d\Omega \tag{5.27}$$

$$\therefore \iiint_{\Omega} \left[ D \underline{\underline{B}}^T \underline{\underline{I}}\, \underline{\underline{B}} - \mu_a \underline{N}\, \underline{N}^T \right] d\Omega \underline{\phi} = \iiint_{\Omega} q_0 \underline{N} d\Omega \tag{5.28}$$

Nota-se que todos os termos dessa equação são conhecidos, exceto por $\underline{\phi}$, que é a incógnita do problema. Logo, para determiná-la, basta calcular as integrais de volume acima.

## 5.4 Resolução do problema direto: cálculo das integrais

Nessa seção, calcular-se-á a integral de volume do lado esquerdo da equação acima. O cálculo da integral do lado direito não é necessário, pois não existem fontes de luz internas ao domínio do problema.

O cálculo será feito separadamente para cada termo, começando pelo primeiro:

$$\iiint_{\Omega} D\underline{\underline{B}}^T \underline{\underline{I}}\,\underline{\underline{B}} d\Omega = \iiint_{\Omega} D\frac{1}{4A^2}\begin{bmatrix} \beta_i & \gamma_i \\ \beta_j & \gamma_j \\ \beta_m & \gamma_m \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \beta_i & \beta_j & \beta_m \\ \gamma_i & \gamma_j & \gamma_m \end{bmatrix} d\Omega \quad \textbf{(5.29)}$$

$$\iiint_{\Omega} D\underline{\underline{B}}^T \underline{\underline{I}}\,\underline{\underline{B}} d\Omega = \frac{D}{4A^2}\begin{bmatrix} \beta_i & \gamma_i \\ \beta_j & \gamma_j \\ \beta_m & \gamma_m \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \beta_i & \beta_j & \beta_m \\ \gamma_i & \gamma_j & \gamma_m \end{bmatrix}\iiint_{\Omega} d\Omega \quad \textbf{(5.30)}$$

$$\iiint_{\Omega} D\underline{\underline{B}}^T \underline{\underline{I}}\,\underline{\underline{B}} d\Omega = \frac{D}{4A^2}\begin{bmatrix} \beta_i^2+\gamma_i^2 & \beta_i\beta_j+\gamma_i\gamma_j & \beta_i\beta_m+\gamma_i\gamma_m \\ \beta_j\beta_i+\gamma_j\gamma_i & \beta_j^2+\gamma_j^2 & \beta_j\beta_m+\gamma_j\gamma_m \\ \beta_m\beta_i+\gamma_m\gamma_i & \beta_m\beta_j+\gamma_m\gamma_j & \beta_m^2+\gamma_m^2 \end{bmatrix} At \quad \textbf{(5.31)}$$

Nessa expressão, $t$ é a espessura do elemento finito.

$$\therefore \iiint_{\Omega} D\underline{\underline{B}}^T \underline{\underline{I}}\,\underline{\underline{B}} d\Omega = \frac{Dt}{4A}\underline{\underline{M}} \quad \textbf{(5.32)}$$

Nessa expressão, $\underline{\underline{M}} = M_{pq} = \beta_p\beta_q + \gamma_p\gamma_q$.

Agora, será analisado o segundo termo:

$$\iiint_{\Omega} \mu_a \underline{N}\,\underline{N}^T d\Omega = \iiint_{\Omega} \mu_a \begin{bmatrix} N_i \\ N_j \\ N_m \end{bmatrix}\begin{bmatrix} N_i & N_j & N_m \end{bmatrix} d\Omega \quad \textbf{(5.33)}$$

$$\iiint_{\Omega} \mu_a \underline{N}\,\underline{N}^T d\Omega = \mu_a \iiint_{\Omega}\begin{bmatrix} N_i^2 & N_iN_j & N_iN_m \\ N_jN_i & N_j^2 & N_jN_m \\ N_mN_i & N_mN_j & N_m^2 \end{bmatrix} d\Omega = \mu_a \iiint_{\Omega} \underline{\underline{P}} d\Omega \quad \textbf{(5.34)}$$

As funções $N_k = \frac{1}{2A}(\alpha_k + \beta_k x + \gamma_k y), \forall k \in \{i, j, m\}$ variam com a posição. Logo, não podem ser retiradas da integral. Calcula-se, então, a integral para um elemento genérico da matriz $\underline{\underline{P}}$:

$$\iiint_{\Omega} P_{pq} d\Omega = \iiint_{\Omega} N_p N_q d\Omega \quad \textbf{(5.35)}$$

$$\iiint_\Omega P_{pq} d\Omega = \frac{1}{4A^2} \iiint_\Omega (\alpha_p + \beta_p x + \gamma_p y)(\alpha_q + \beta_q x + \gamma_q y) d\Omega \qquad \textbf{(5.36)}$$

$$\iiint_\Omega P_{pq} d\Omega = \frac{1}{4A^2} \iiint_\Omega \left[\alpha_p \alpha_q + \beta_p \beta_q x^2 + \gamma_p \gamma_q y^2 + (\alpha_p \beta_q + \alpha_q \beta_p) x + \ldots \right.$$
$$\left. \ldots + (\alpha_p \gamma_q + \alpha_q \gamma_p) y + (\beta_p \gamma_q + \beta_q \gamma_p) xy \right] d\Omega \qquad \textbf{(5.37)}$$

Sem perda de generalidade, pode-se considerar que a origem das coordenadas está no centróide do elemento finito. Com essa hipótese adicional, a integral é mais facilmente calculada[8]:

$$\iiint_\Omega P_{pq} d\Omega = \frac{t}{4A} \left[ \alpha_p \alpha_q + \beta_p \beta_q \frac{x_i^2 + x_j^2 + x_m^2}{12} + \gamma_p \gamma_q \frac{y_i^2 + y_j^2 + y_m^2}{12} + \ldots \right.$$
$$\left. \ldots + (\beta_p \gamma_q + \beta_q \gamma_p) \frac{x_i y_i + x_j y_j + x_m y_m}{12} \right] \qquad \textbf{(5.38)}$$

Aplicando na expressão original:

$$\iiint_\Omega \mu_a \underline{N}\,\underline{N}^T d\Omega = \mu_a \iiint_\Omega \underline{\underline{P}} d\Omega = \frac{\mu_a t}{4A} \underline{\underline{H}} \qquad \textbf{(5.39)}$$

Nessa expressão:

$$\underline{\underline{H}} = H_{pq} = \alpha_p \alpha_q + \beta_p \beta_q \frac{x_i^2 + x_j^2 + x_m^2}{12} + \gamma_p \gamma_q \frac{y_i^2 + y_j^2 + y_m^2}{12} + \ldots$$
$$\ldots + (\beta_p \gamma_q + \beta_q \gamma_p) \frac{x_i y_i + x_j y_j + x_m y_m}{12} \qquad \textbf{(5.40)}$$

Assim, fica definido o segundo termo da integral.

Resumindo as fórmulas acima, o problema direto é escrito assim:

$$\left[ \frac{Dt}{4A} \underline{\underline{M}} - \frac{\mu_a t}{4A} \underline{\underline{H}} \right] \underline{\phi} = \iiint_\Omega q_0 \underline{N}^T d\Omega \qquad \textbf{(5.41)}$$

## 5.5 Resolução do problema inverso: algoritmo Caixa-Preta

Esse algoritmo consiste na obtenção de um mapeamento linear que relaciona perturbações de absorção de luz com as medidas nos fotodiodos. Mais precisamente, deseja-se estimar uma matriz $B$ tal que:

$$\Theta = B\Lambda \tag{5.42}$$

Nessa expressão:

- "$\Theta$" é uma matriz de perturbações na distribuição de absorção $\mu_a(r)$ (e conseqüentemente também na distribuição da difusão $D(r)$). Levando em conta a existência de $n$ elementos finitos nos quais o domínio foi discretizado, $\Theta$ será uma matriz $nxn$. O elemento $\Theta_{ij}$ será uma perturbação arbitrária (não-nula) da absorção no i-ésimo elemento finito se $i = j$, e zero se $i \neq j$. Isto é, será uma matriz diagonal.
- "$\Lambda$" é uma matriz de perturbações na distribuição de densidade de fótons $\Phi$, captada pelos fotodiodos. Levando em conta a existência de $p$ emissores de luz, é possível realizar $p$ experimentos diferentes, ao se acender um e apenas um LED por vez. Adicionalmente, considerando a existência de $q$ receptores, cada experimento gera $q$ medidas dos fotodiodos. Dessa forma, para cada perturbação $\Theta_k$ (a k-ésima coluna de $\Theta$), geram-se $pq$ perturbações. Dessa forma, $\Lambda$ é uma matriz $pqxn$.

Obviamente, essas perturbações são medidas em relação a um estado homogêneo, no qual a distribuição de absorção $\mu_a(r)$ é constante em todo o domínio, e tem valor $\mu_0$. Através da solução do problema direto, determinam-se, para cada um dos $p$ LED's ligados, $q$ medidas nos fotodiodos. Ou seja, determina-se o vetor $\Lambda_0$, de dimensão $pqx1$, das leituras nos fotodiodos sem perturbação.

Resolvendo o problema direto com a perturbações de absorção no k-ésimo elemento finito (ou seja, referente à k-ésima coluna de $\Theta$), obtém-se um vetor $\Lambda_k$. Então calcula-se a k-ésima coluna de $\Lambda$ como sendo a diferença $\Lambda_k - \Lambda_0$.

Finalmente, para se obter $B$, basta isolá-la na expressão inicial. Como a matriz $\Lambda$, em geral, não é quadrada, é necessário multiplicá-la por sua transposta. Dessa forma:

$$\Theta\Lambda^T = B\Lambda\Lambda^T \tag{5.43}$$

$$\Theta\Lambda^T\left(\Lambda\Lambda^T\right)^{-1} = B \tag{5.44}$$

É possível que a matriz $\left(\Lambda\Lambda^t\right)$ tenha determinante nulo (ou muito próximo de zero). Sob essa condição, sua inversa não estará definida (ou enfrentará problemas computacionais em seu cálculo). Para contornar esse problema, utiliza-se uma função de regularização na forma de uma matriz diagonal, a ver:

$$\Theta\Lambda^T\left(\Lambda\Lambda^T + \alpha I\right)^{-1} = B \tag{5.45}$$

Assim, garante-se que a inversa está definida, e a matriz $B$ pode ser encontrada.

Mostrou-se, assim, como proceder para obter uma matriz $B$ a partir de resoluções sucessivas do problema direto. Com essa matriz em mãos, pode-se empregá-la para estimar perturbações de difusão no domínio $(\Theta)$ a partir de medidas nos fotodiodos $(\Lambda)$. Ou seja, com a matriz $B$ em mãos, é possível reconstruir a imagem (formada pela distribuição de difusão) a partir das medidas dos fotodiodos, atingindo assim o objetivo do algoritmo.

Um programa em C foi escrito para realizar as tarefas descritas nesse capítulo. Seu código-fonte encontra-se no ANEXO B.

# 6. *SOFTWARE*: RESULTADOS EXPERIMENTAIS (TESTES-FANTASMA)

Três passos principais compõem os testes experimentais no *software*: a geração da malha de elementos finitos, a solução do problema direto, e a solução do problema inverso. Cada uma delas será detalhada nesse capítulo.

## 6.1 Geração da malha de elementos finitos

A malha de elementos finitos utilizada como dado de entrada para o programa deve cumprir três condições primordiais:

- Seu contorno externo deve ser circular, pois esse é o formato da cuba na qual o *hardware* está montado.
- Esse contorno deve ter o mesmo diâmetro da cuba mencionada.
- Ela deve conter seis nós eqüidistantes em seu contorno, pois a matriz de LED's e fotodiodos (que está no contorno da cuba) tem esse número de elementos.

Com essas premissas em mente, o *software* "gmesh" foi empregado para se obter a seguinte malha de elementos finitos:

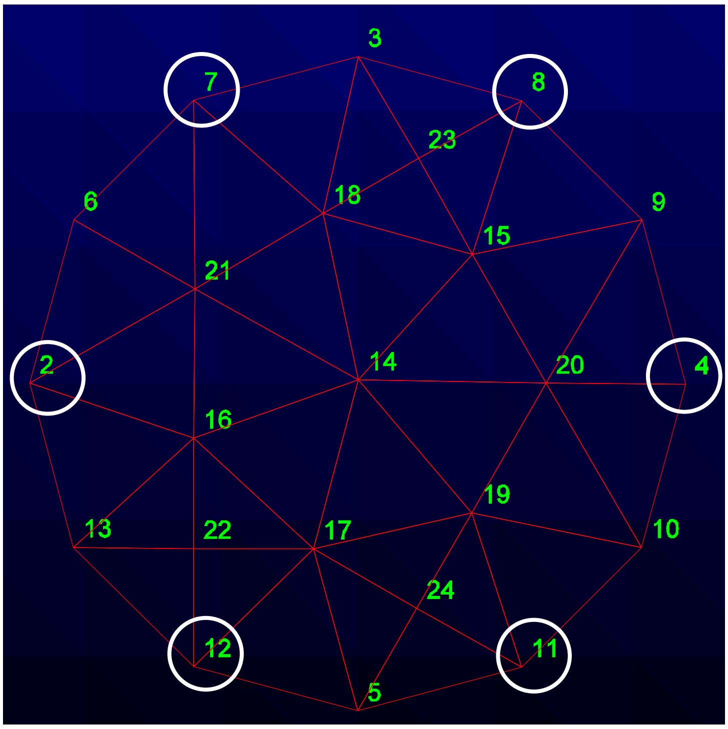



**Ilustração 16: Malha de elementos finitos, com indicação dos números dos nós, e com os nós onde se encontram os pares de LED e foto-diodo destacados.**

Como se pode verificar, a Ilustração 16 mostra inclusive os números dos nós que compõem a malha. Os nós circulados indicam a posição dos pares de emissão e recepção de raios infravermelhos. As condições primordiais enumeradas acima foram cumpridas, pois, além de o contorno ser circular e os nós destacados serem eqüidistantes, o diâmetro do círculo é de 10cm, o que condiz com a dimensão do protótipo.

Existem 23 nós na malha, que foram numerados pelo programa de 2 a 24. Essa numeração é corrigida internamente no programa (escrito pelo autor), de forma que ela comece no 1 e termine no 23 (vide ANEXO B).

Além da ilustração, o "gmesh" também gera uma tabela com as coordenadas de cada um dos nós. Essa tabela foi gravada no arquivo "coord.dat", o qual é lido pelo programa (vide ANEXO B). Segue a listagem desse arquivo na Tabela 9:

**Tabela 9: Listagem dos nós da malha ("coord.dat").**

| Nó | Coordenada x | Coordenada y | Dígito de fim de linha |
|---|---|---|---|
| 2 | -0.05 | 0 | 0 |
| 3 | 0 | 0.05 | 0 |
| 4 | 0.05 | 0 | 0 |
| 5 | 0 | -0.05 | 0 |
| 6 | -0.0433012701892233 | 0.0249999999999932 | 0 |
| 7 | -0.0250000000000106 | 0.0433012701892133 | 0 |
| 8 | 0.0249999999999932 | 0.0433012701892233 | 0 |
| 9 | 0.0433012701892133 | 0.0250000000000106 | 0 |
| 10 | 0.0433012701892233 | -0.0249999999999932 | 0 |
| 11 | 0.0250000000000106 | -0.0433012701892133 | 0 |
| 12 | -0.0249999999999932 | -0.0433012701892233 | 0 |
| 13 | -0.0433012701892133 | -0.0250000000000106 | 0 |
| 14 | 0.000157939560455404 | 0.000558875067241927 | 0 |
| 15 | 0.0174736078779807 | 0.019822108365396 | 0 |
| 16 | -0.024980620707297 | -0.00833943635561119 | 0 |
| 17 | -0.00672523974077480 | -0.0252478481461462 | 0 |
| 18 | -0.00528006564578330 | 0.0260248707464720 | 0 |
| 19 | 0.0174697682008788 | -0.0197417787610027 | 0 |
| 20 | 0.0287466701397078 | 5.58373173819647e-05 | 0 |
| 21 | -0.0248625117366571 | 0.0144339498169099 | 0 |
| 22 | -0.0249953174063598 | -0.0252280369980400 | 0 |
| 23 | 0.00922753041277986 | 0.0344695463252280 | 0 |
| 24 | 0.00904870977195821 | -0.0343583687770560 | 0 |

Além da enumeração dos nós, o "gmesh" também enumera os próprios elementos finitos, como mostra a Ilustração 17:

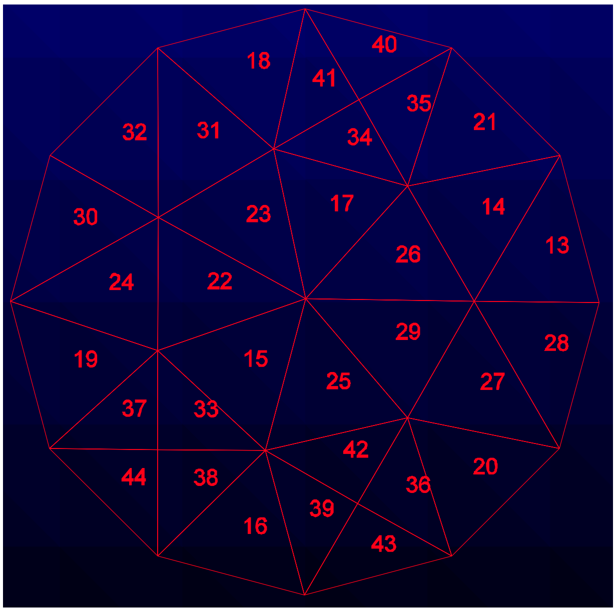



**Ilustração 17: Malha de elementos finitos, com indicação dos números dos elementos.**

Existem 32 elementos na malha, que foram numerados pelo programa de 13 a 44. Essa numeração é corrigida internamente no programa (escrito pelo autor), de forma que ela comece no 1 e termine no 32 (vide ANEXO B).

Além da ilustração, o "gmesh" também gera uma tabela indicando os nós que compõem cada um dos elementos. Essa tabela foi gravada no arquivo "topo.dat", o qual é lido pelo programa (vide ANEXO B). Segue a listagem desse arquivo na

**Tabela 10: Listagem da topologia da malha ("topo.dat").**

| Elemento | Informação para o caso 3D (não utilizada) | | | | Nó i | Nó j | Nó m |
|---|---|---|---|---|---|---|---|
| 13 | 2 | 1 | 1 | 3 | 9 | 4 | 20 |
| 14 | 2 | 1 | 1 | 3 | 15 | 9 | 20 |
| 15 | 2 | 1 | 1 | 3 | 16 | 14 | 17 |
| 16 | 2 | 1 | 1 | 3 | 5 | 12 | 17 |
| 17 | 2 | 1 | 1 | 3 | 15 | 14 | 18 |
| 18 | 2 | 1 | 1 | 3 | 7 | 3 | 18 |
| 19 | 2 | 1 | 1 | 3 | 13 | 2 | 16 |
| 20 | 2 | 1 | 1 | 3 | 10 | 11 | 19 |
| 21 | 2 | 1 | 1 | 3 | 8 | 9 | 15 |
| 22 | 2 | 1 | 1 | 3 | 14 | 16 | 21 |
| 23 | 2 | 1 | 1 | 3 | 18 | 14 | 21 |
| 24 | 2 | 1 | 1 | 3 | 16 | 2 | 21 |
| 25 | 2 | 1 | 1 | 3 | 17 | 14 | 19 |
| 26 | 2 | 1 | 1 | 3 | 14 | 15 | 20 |
| 27 | 2 | 1 | 1 | 3 | 10 | 19 | 20 |
| 28 | 2 | 1 | 1 | 3 | 4 | 10 | 20 |
| 29 | 2 | 1 | 1 | 3 | 19 | 14 | 20 |
| 30 | 2 | 1 | 1 | 3 | 2 | 6 | 21 |
| 31 | 2 | 1 | 1 | 3 | 7 | 18 | 21 |
| 32 | 2 | 1 | 1 | 3 | 6 | 7 | 21 |
| 33 | 2 | 1 | 1 | 3 | 16 | 17 | 22 |
| 34 | 2 | 1 | 1 | 3 | 15 | 18 | 23 |
| 35 | 2 | 1 | 1 | 3 | 8 | 15 | 23 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 36 | 2 | 1 | 1 | 3 | 19 | 11 | 24 |
| 37 | 2 | 1 | 1 | 3 | 13 | 16 | 22 |
| 38 | 2 | 1 | 1 | 3 | 17 | 12 | 22 |
| 39 | 2 | 1 | 1 | 3 | 5 | 17 | 24 |
| 40 | 2 | 1 | 1 | 3 | 3 | 8 | 23 |
| 41 | 2 | 1 | 1 | 3 | 18 | 3 | 23 |
| 42 | 2 | 1 | 1 | 3 | 17 | 19 | 24 |
| 43 | 2 | 1 | 1 | 3 | 11 | 5 | 24 |
| 44 | 2 | 1 | 1 | 3 | 12 | 13 | 22 |

Uma informação geométrica adicional que foi empregada na solução é a espessura da malha. Como se pôde verificar, ela é bidimensional, e o problema todo é tratado como bidimensional. Entretanto, a espessura aparece nas equações matemáticas que descrevem o fenômeno físico de difusão de fótons, de forma que se torna necessário defini-la. A espessura utilizada foi de 3cm. Esse valor corresponde experimentalmente à espessura na qual a luz se difunde notavelmente na direção perpendicular ao plano da matriz dos LED's e fotodiodos.

Com as informações exibidas acima, a malha de elementos finitos fica completamente definida, e pode ser utilizada pelo programa nos passos a seguir.

## 6.2 **Solução do problema direto**

Além das informações geométricas já definidas, é necessário definir as informações de material para se resolver o problema direto. Isto é, precisa-se definir o coeficiente de absorção $\mu_a$ e de dispersão reduzido $\mu_s'$. O meio empregado na experiência é uma mistura de água e leite. Nas pesquisas bibliográficas, não se encontrou valores dos coeficientes para esse material. Logo, ele teve que ser determinado experimentalmente, buscando-se os valores que implicavam em uma resposta que mais se aproximava da realidade, do ponto de vista qualitativo. Os valores encontrados foram: $\mu_a = 10m^{-1}$ e $\mu_s' = 30m^{-1}$.

Vale lembrar que o valor de $\mu_s'$ permanecerá constante em todas as simulações, sendo que apenas o valor de $\mu_a$ vai variar.

Com o programa, as informações geométricas e de material em mãos, é possível determinar a resposta do sistema a uma excitação. Por resposta do sistema, entenda-se a distribuição de $\Phi$, a densidade de fótons, e por excitação entenda-se um (ou mais) LED ligado.

Em primeiro lugar, determinou-se o comportamento do sistema com $\mu_a = 10m^{-1}$ constante, e apenas um LED ligado. A distribuição de densidade de fótons pode ser vista na Ilustração 18:

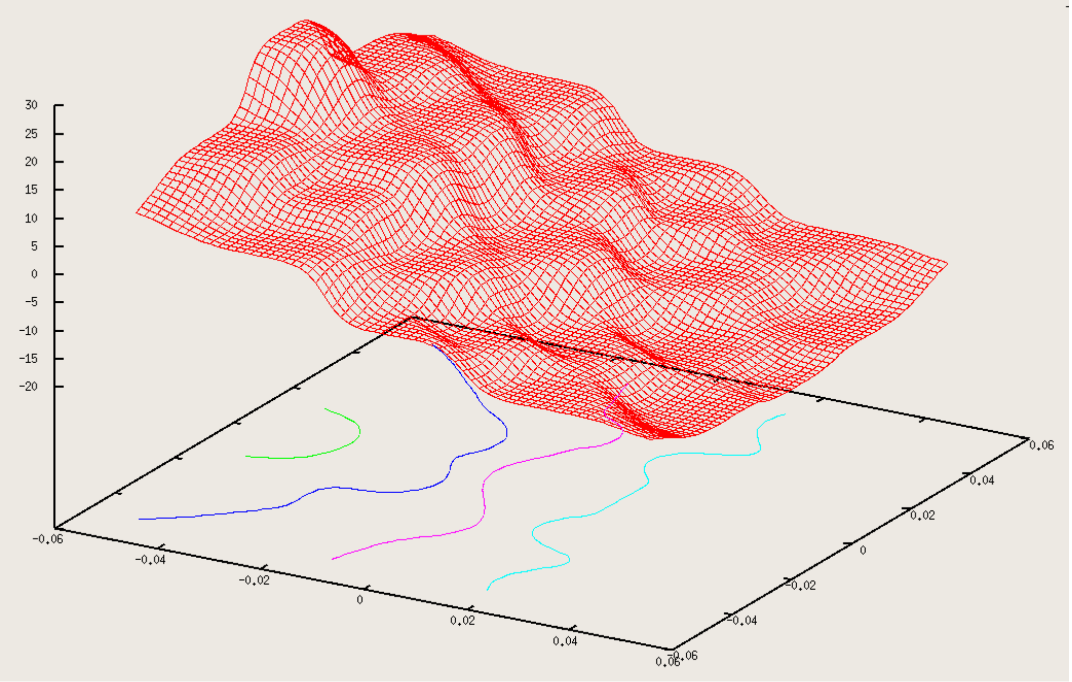



**Ilustração 18: Problema direto não perturbado, com apenas um LED (resposta 3D).**

O LED ligado encontra-se à esquerda da figura, onde se verifica um pico de densidade de fótons. Conforme se afasta dessa fonte, a densidade de fótons cai, segundo um padrão aproximadamente circular, como era de se esperar.

Nota: para gerar essa e as próximas figuras, empregou-se o *software* "gnuplot". Todos os gráficos estão desenhados sobre uma malha retangular fictícia de 64x64. Mas os valores correspondem àqueles da malha apresentada acima, que tem contorno circular.

A Ilustração 19 mostra, ao mesmo tempo, a malha de elementos finitos (à esquerda), indicando qual é o LED que está aceso, e a distribuição de densidade de fótons através das linhas de mesma densidade (à direita):

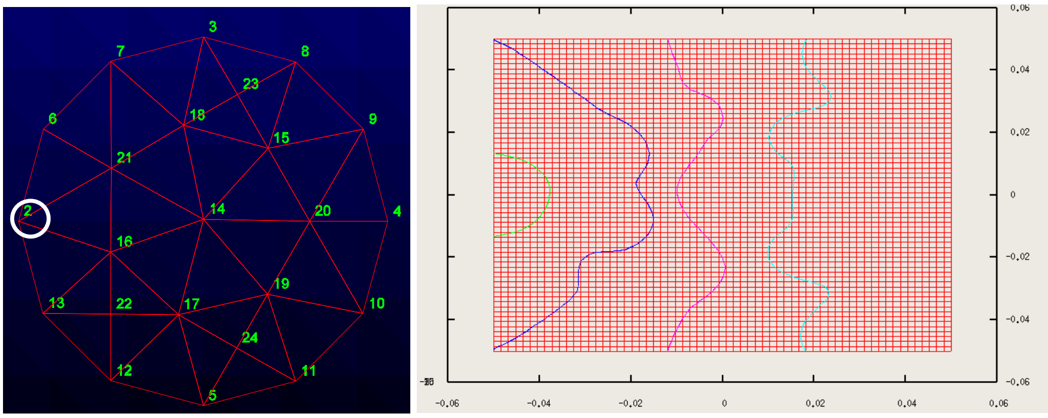


**Ilustração 19: Problema direto não perturbado, com apenas um LED (resposta 2D).**

Um segundo teste consistiu em determinar o comportamento do sistema ainda com $\mu_a = 10m^{-1}$ constante, mas agora com todos os LED's ligados. A distribuição de densidade de fótons pode ser vista na Ilustração 20:

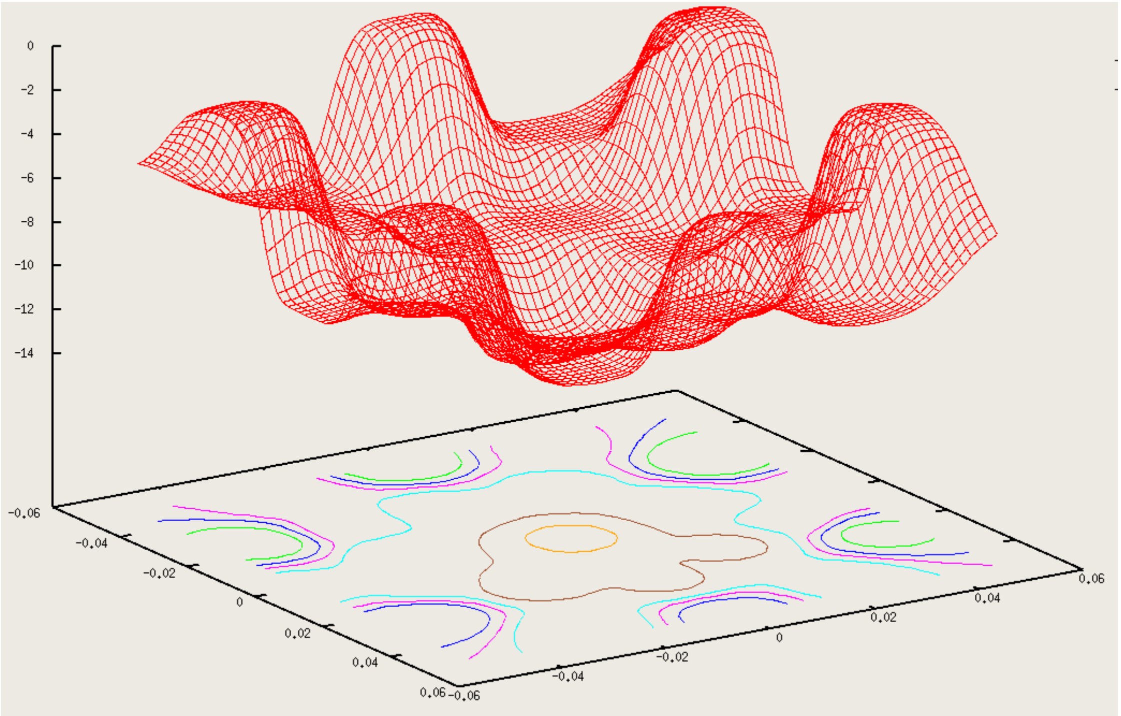


**Ilustração 20: Problema direto não perturbado, com todos os LED's (resposta 3D).**

Como se pode verificar, existem picos de densidade de fótons próximo a todos os LED's, e a mesma cai conforme se aproxima do centro da malha.

A Ilustração 21 mostra, ao mesmo tempo, a malha de elementos finitos (à esquerda), indicando que todos os LED's estão acesos, e a distribuição de densidade de fótons através das linhas de mesma densidade (à direita):

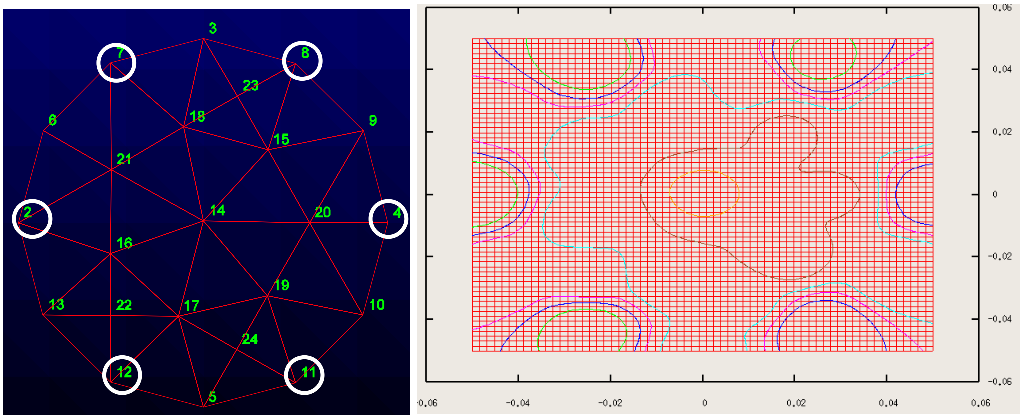



**Ilustração 21: Problema direto não perturbado, com todos os LED's (resposta 2D).**

O terceiro teste consistiu em determinar o comportamento do sistema perturbado. Isto é, com uma região na qual $\mu_a = 15m^{-1}$. Essa região está delimitada pelas circunferências verdes, na Ilustração 22:

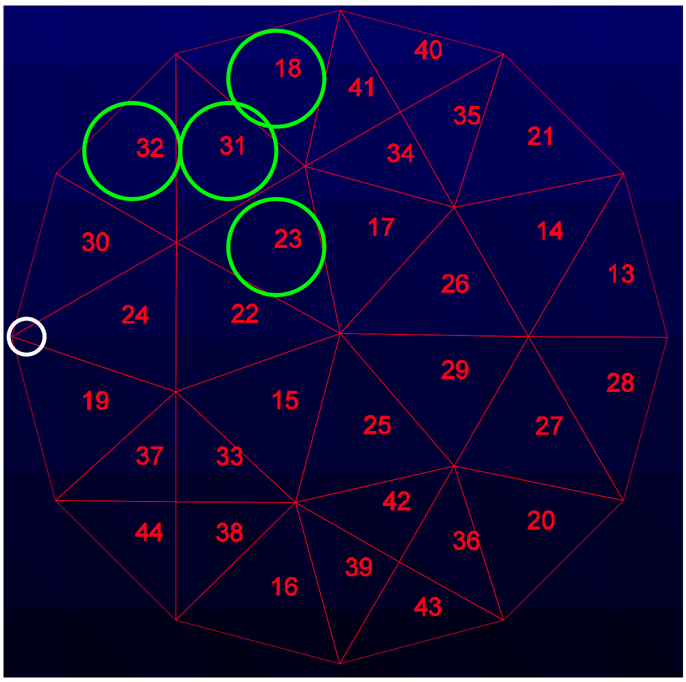



**Ilustração 22: Malha de elementos finitos perturbada por absorção maior que o normal (problema direto).**

O círculo branco indica o único LED que será ligado. A distribuição de densidade de fótons pode ser vista na Ilustração 23:

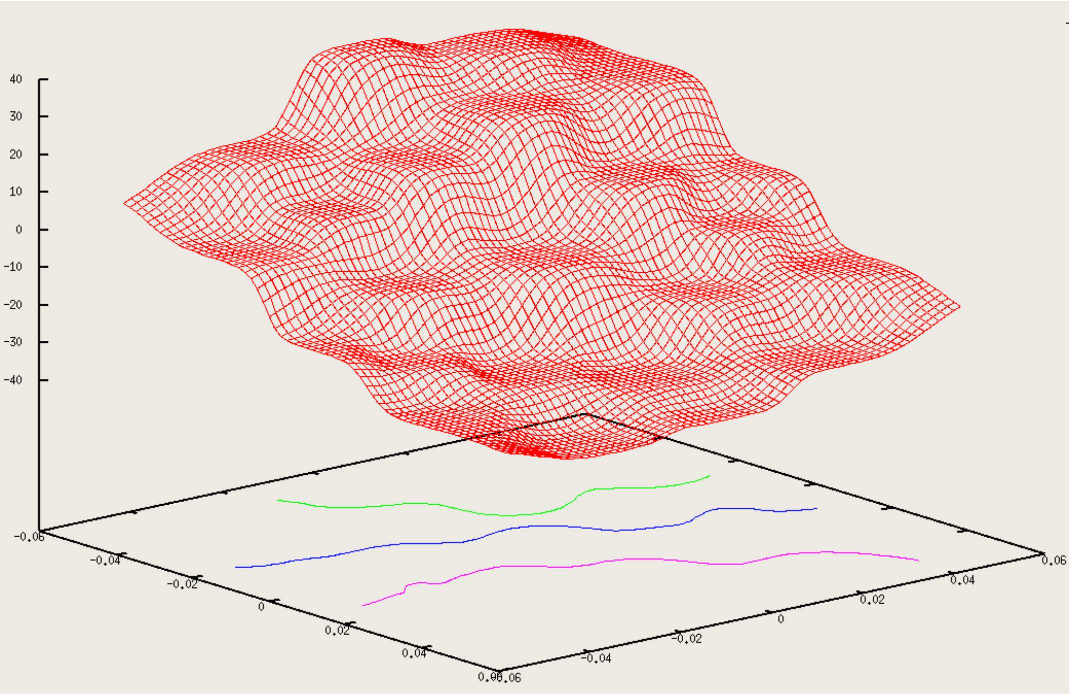


**Ilustração 23: Problema direto perturbado (absorção maior), com apenas um LED (resposta 3D).**

Como se pode verificar, a densidade de fótons é maior na região perturbada do que no restante do domínio. Era exatamente o que se esperava, uma vez que a maior absorção dessa região induz uma menor dispersão dos fótons para as regiões adjacentes. Como conseqüência, os fótons ficam retidos na região perturbada.

A Ilustração 24 mostra, ao mesmo tempo, a malha de elementos finitos (à esquerda), indicando o único LED aceso, e a distribuição de densidade de fótons através das linhas de mesma densidade (à direita):

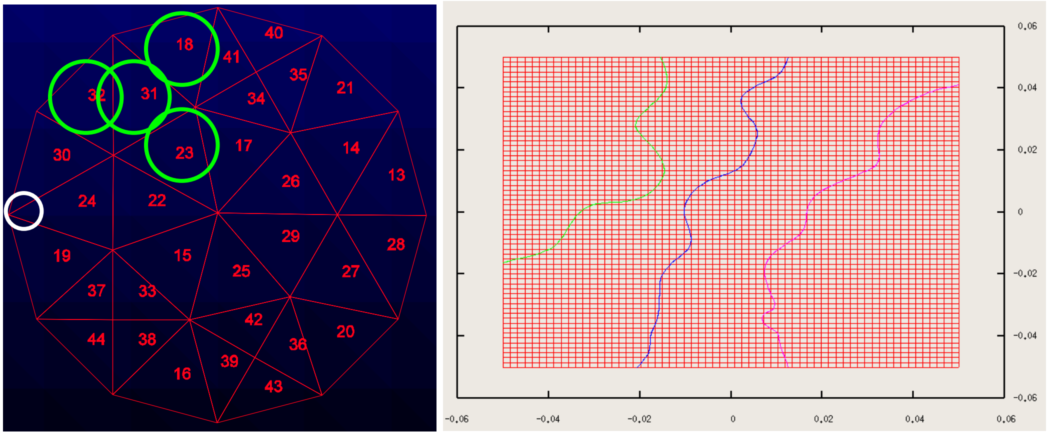


**Ilustração 24: Problema direto perturbado (absorção maior), com apenas um LED (resposta 2D).**

O quarto teste consistiu em determinar o comportamento do sistema perturbado de forma oposta ao terceiro. Isto é, com uma região na qual $\mu_a = 0$. Essa região está delimitada pelas circunferências verdes, na Ilustração 25:

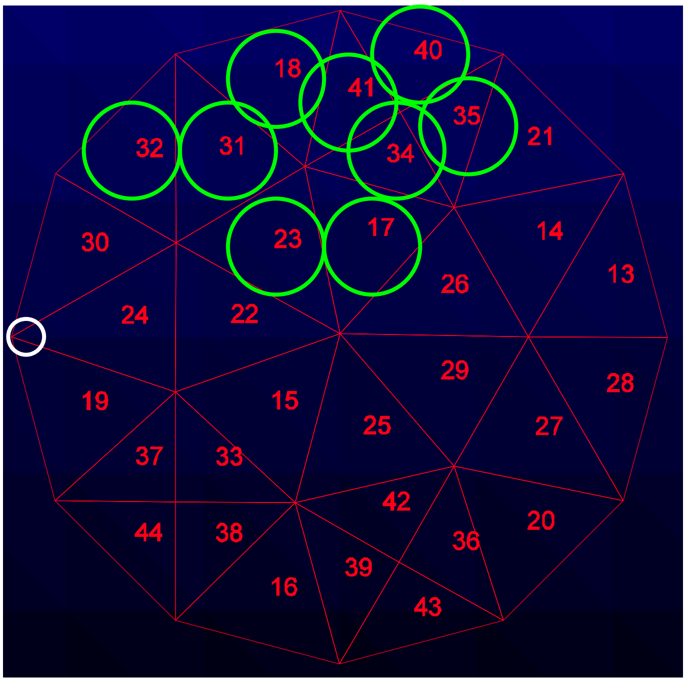



**Ilustração 25: Malha de elementos finitos perturbada por absorção menor que o normal (problema direto).**

O círculo branco indica o único LED que será ligado. A distribuição de densidade de fótons pode ser vista na Ilustração 26:

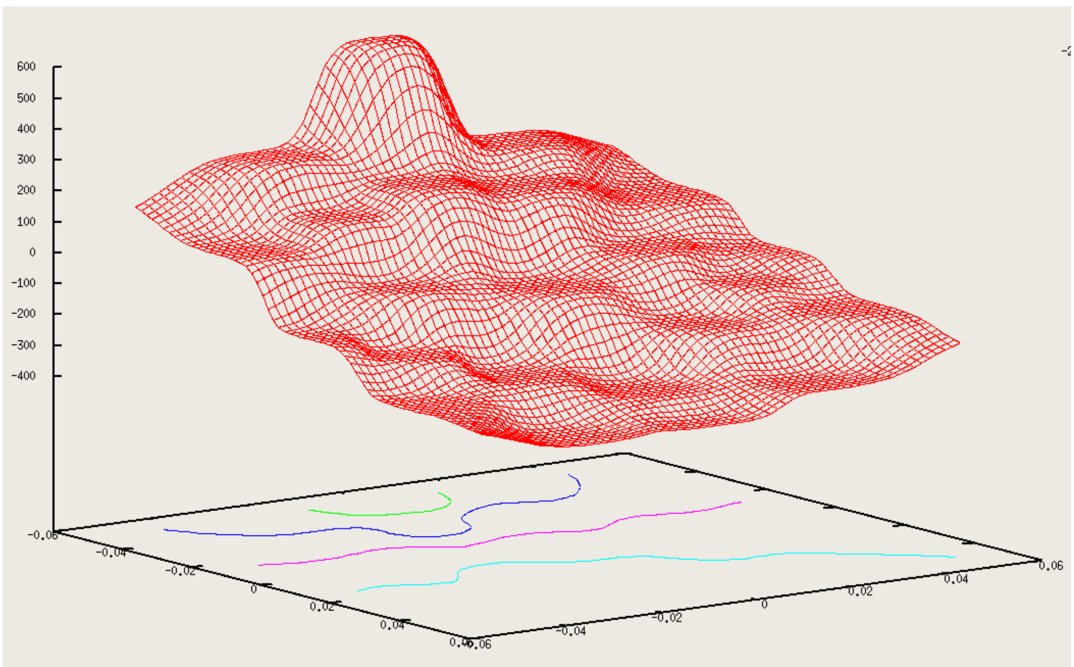



**Ilustração 26: Problema direto perturbado (absorção menor), com apenas um LED (resposta 3D).**

Como se pode verificar, a redução na densidade de fótons é menos intensa na região perturbada do que no restante do domínio. Era exatamente o que se esperava, uma vez que a menor absorção dessa região induz uma maior dispersão dos fótons para as regiões adjacentes. Como conseqüência, os fótons conseguem atravessar com mais facilidade a região perturbada.

A Ilustração 27 mostra, ao mesmo tempo, a malha de elementos finitos (à esquerda), indicando o único LED aceso, e a distribuição de densidade de fótons através das linhas de mesma densidade (à direita):

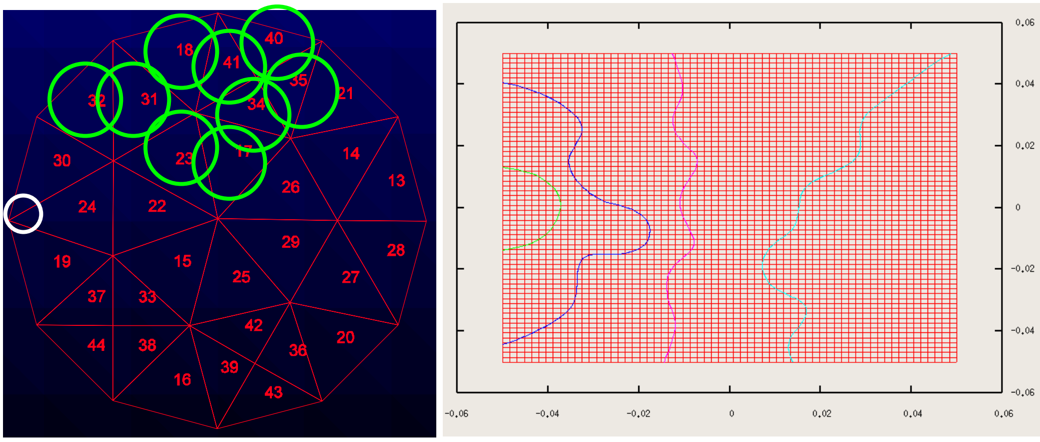



**Ilustração 27: Problema direto perturbado (absorção menor), com apenas um LED (resposta 2D).**

Como se verificou, a execução do programa está gerando distribuições plausíveis para o problema direto de difusão de luz em meio translúcido. Logo, ele é confiável para ser empregado na próxima seção, no problema inverso.

## 6.3 Solução do problema inverso

Para se testar a eficácia do algoritmo de Caixa-Preta para se resolver o problema inverso, os testes-fantasma foram empregados. Esses testes consistem em impor uma perturbação arbitrária à distribuição de $\mu_a$ no domínio, gravar as leituras obtidas nos fotodiodos, e em seguida rodar o algoritmo Caixa-Preta para se tentar reconstruir a perturbação.

Vale lembrar que, entre a seção anterior e essa, existe uma etapa dentro do programa, no qual a matriz $B$ é calculada. Como a teoria por trás desse cálculo já foi exposta, e sua implementação pode ser verificada no ANEXO B, não há necessidade de se entrar em maiores detalhes sobre o valor numérico dos elementos de $B$ aqui.

Vale mencionar, entretanto, que o valor do coeficiente de correção empregado foi de apenas $\alpha = 10^{-5}$.

Três dos testes realizados serão aqui detalhados. No primeiro deles, apenas um dos elementos finitos teve o valor de $\mu_a$ modificado para $\mu_a = 15m^{-1}$. Ele está identificado na Ilustração 28:

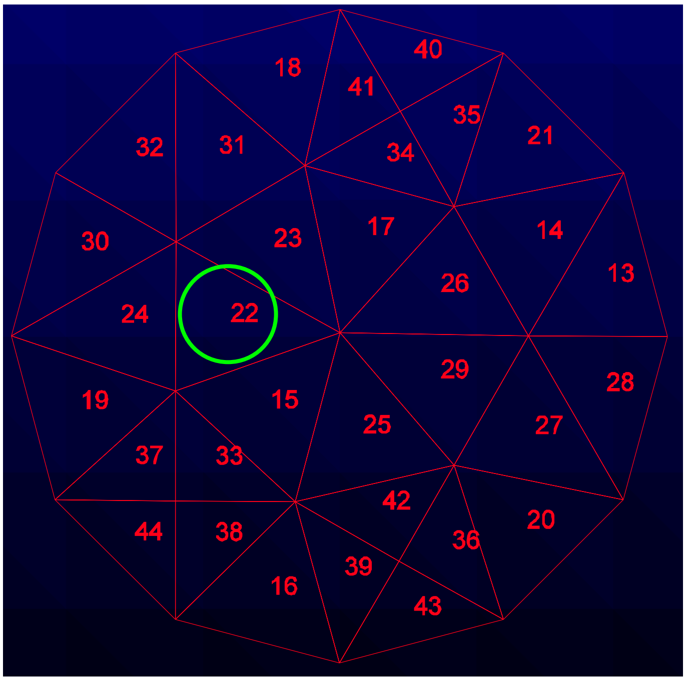



**Ilustração 28: Malha de elementos finitos perturbada (primeiro teste-fantasma).**

A execução do programa gerou o resultado de distribuição de $\mu_a$ exibido na Ilustração 29:

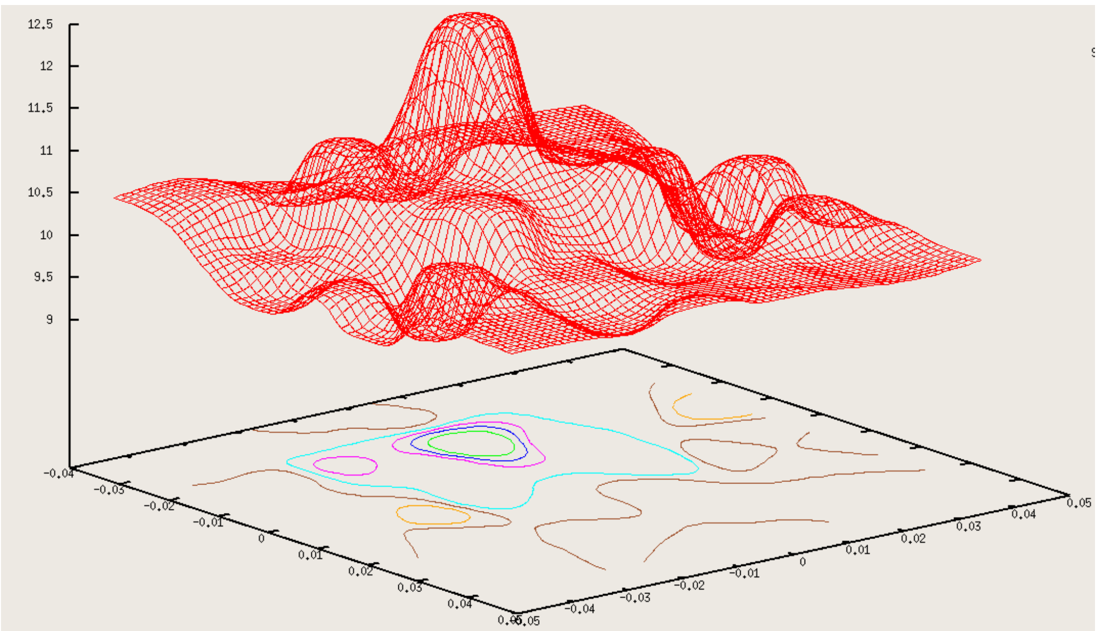



**Ilustração 29: Distribuição de μ (primeiro teste-fantasma, 3D).**

Verifica-se um forte pico na região esperada (onde a perturbação foi introduzida). O resto da distribuição não é plenamente uniforme, mas nota-se que as oscilações fora da região perturbada são de amplitude dez vezes menor que dentro da mesma.

A Ilustração 30 mostra a forte correlação entre a posição e a dimensão da região perturbada na malha e na resposta do programa:

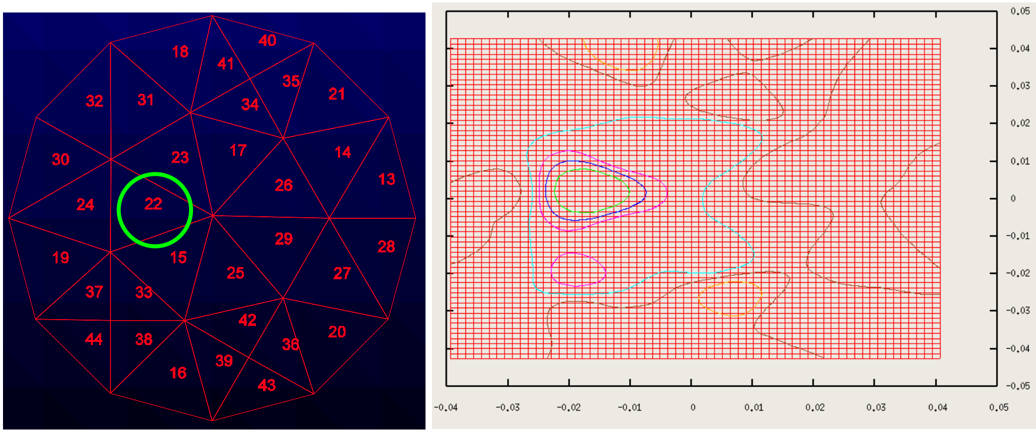



**Ilustração 30: Distribuição de μ (primeiro teste-fantasma, 2D).**

No segundo teste-fantasma, apenas dois elementos finitos tiveram o valor de $\mu_a$ modificado para $\mu_a = 15m^{-1}$. Eles estão identificados na Ilustração 31:

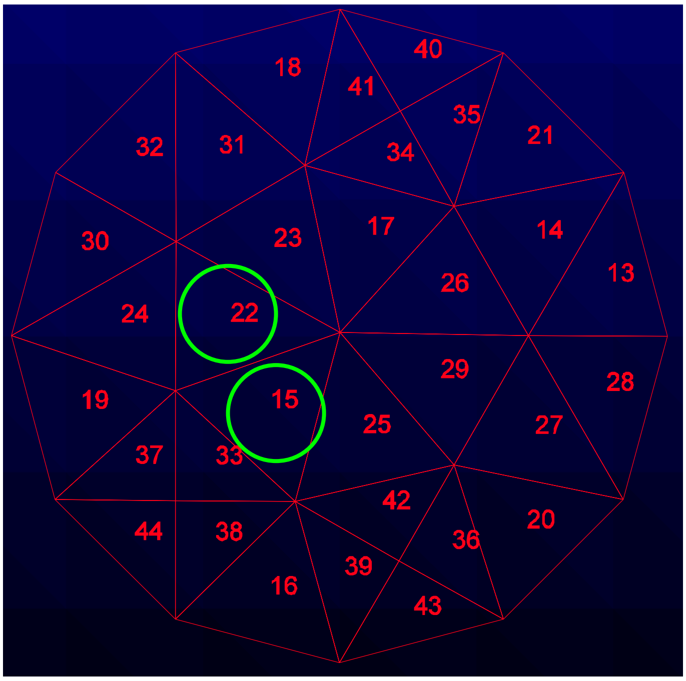



**Ilustração 31: Malha de elementos finitos perturbada (segundo teste-fantasma).**

A execução do programa gerou o resultado de distribuição de $\mu_a$ exibido na Ilustração 32:

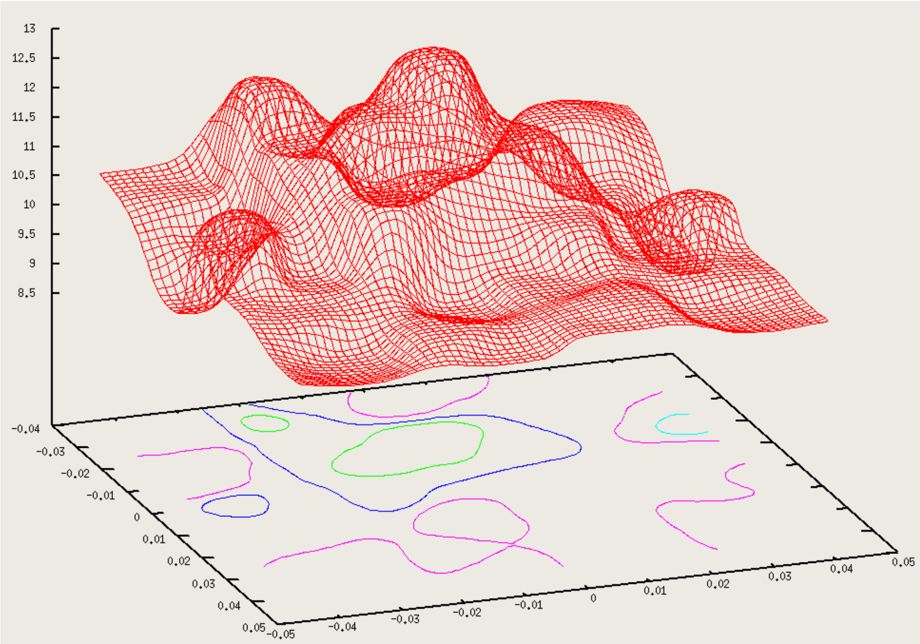



**Ilustração 32: Distribuição de μ (segundo teste-fantasma, 3D).**

Verifica-se um forte pico na região esperada (onde as perturbações foram introduzidas). O resto da distribuição não é plenamente uniforme, mas nota-se que as oscilações fora da região perturbada são de amplitude dez vezes menor que dentro da mesma.

A Ilustração 33 mostra a forte correlação entre a posição e a dimensão da região perturbada na malha e na resposta do programa:

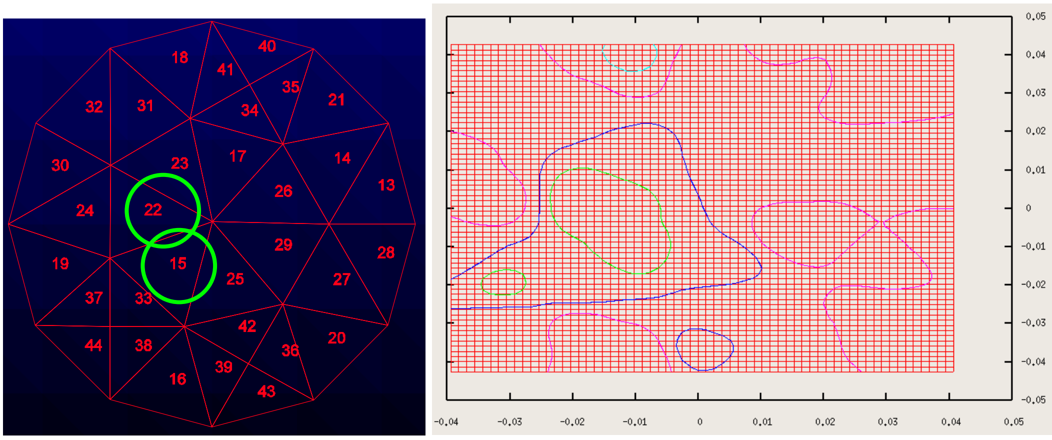



**Ilustração 33: Distribuição de μ (segundo teste-fantasma, 2D).**

No último teste-fantasma, toda a região adjacente a um certo LED teve o valor de $\mu_a$ modificado para $\mu_a = 15m^{-1}$. Essa região está identificada na Ilustração 34:

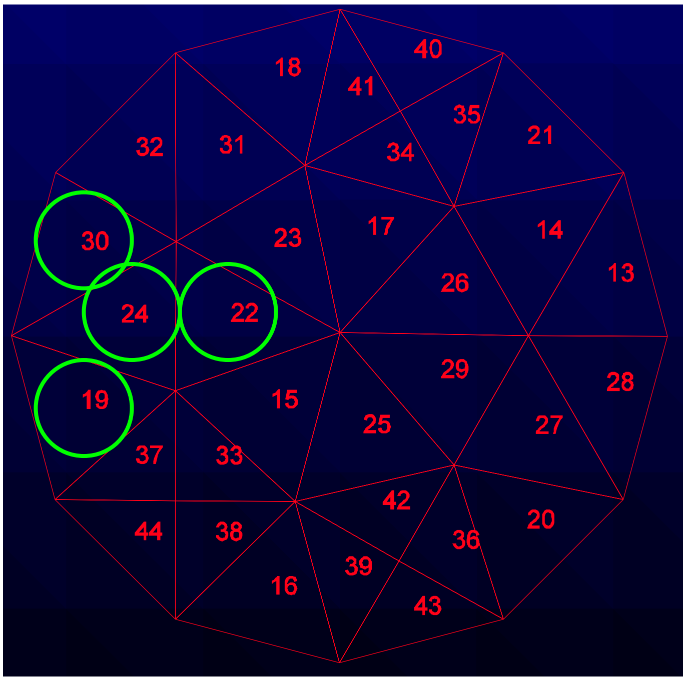



**Ilustração 34: Malha de elementos finitos perturbada (terceiro teste-fantasma).**

A execução do programa gerou o resultado de distribuição de $\mu_a$ exibido na Ilustração 35:

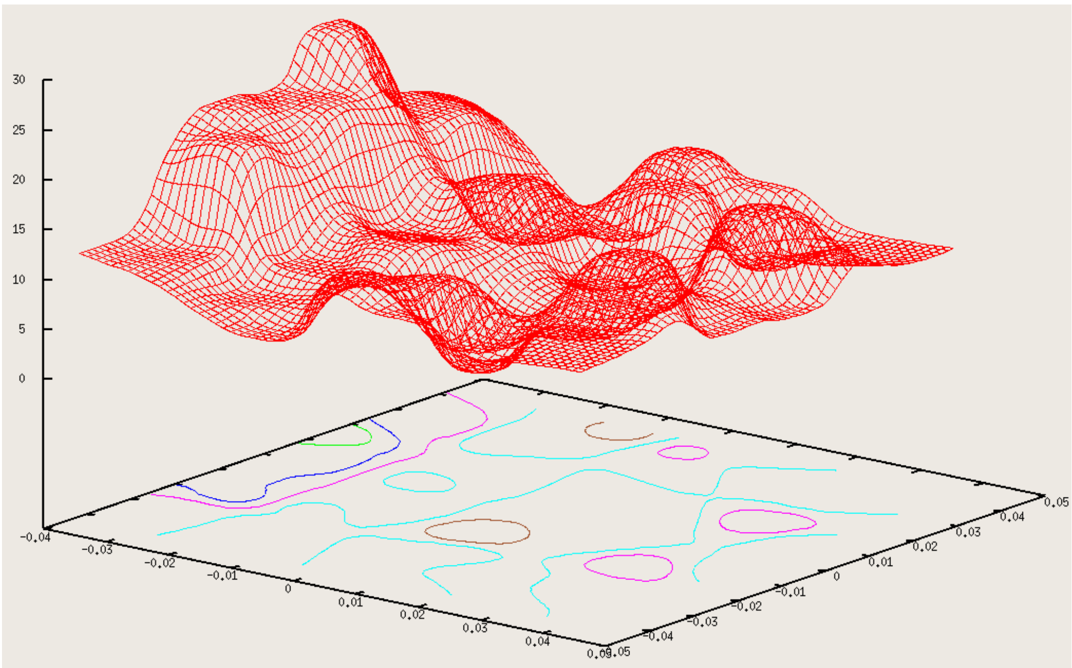


**Ilustração 35: Distribuição de μ (terceiro teste-fantasma, 3D).**

Verifica-se um forte pico na região esperada (onde a perturbação foi introduzida). O resto da distribuição não é plenamente uniforme, e percebe-se que, diferente dos testes anteriores, as oscilações fora da região perturbada já não são de amplitude dez vezes menor que dentro da mesma.

A Ilustração 36 mostra a correlação entre a posição e a dimensão da região perturbada na malha e na resposta do programa:

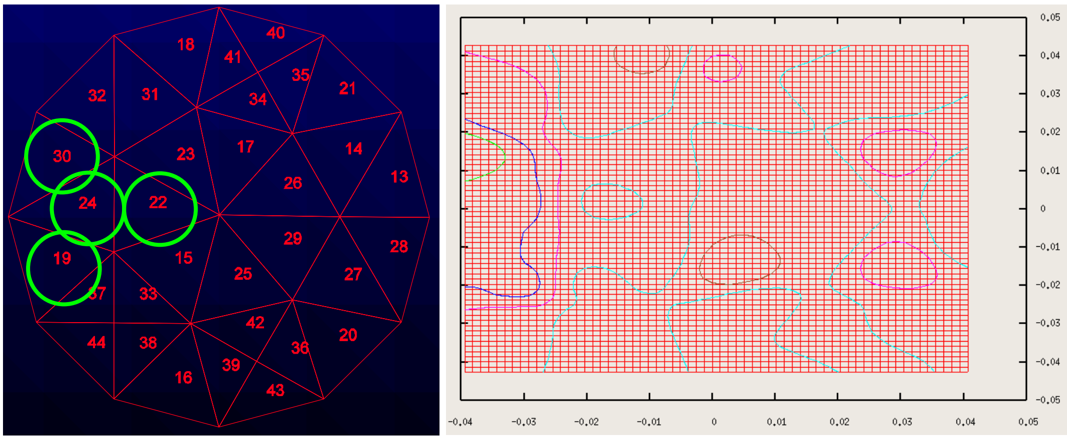



**Ilustração 36: Distribuição de μ (terceiro teste-fantasma, 2D).**

## 6.4 Discussão acerca dos testes-fantasma

Como se verificou nas seções anteriores, tanto o problema direto quanto o inverso foram realizados com sucesso. Como conseqüência, tem-se em mãos uma matriz $B$ confiável para se realizar testes com o protótipo. Mas também vale mencionar aqui as limitações do problema direto e do algoritmo Caixa-Preta.

O problema direto apresentou resultados bastante condizentes com a realidade no caso não perturbado. Mas no caso perturbado, verificou-se uma distribuição bastante distante do padrão circular verificado nos casos anteriores. Essa mudança já foi explicada, mas deve-se ressaltar o quanto o sistema é sensível a mudanças de parâmetros de material: o coeficiente $\mu_a$ foi modificado em apenas quatro elementos (12,5% do total), e a distribuição de densidade de fótons se comportou como se a posição do LED tivesse se deslocado.

Sobre o algoritmo Caixa-Preta, a principal crítica está no fato que, nas regiões onde o coeficiente $\mu_a$ deveria estar constante, ele oscila demasiadamente. Isso se deve a duas razões principais: ao baixo refinamento da malha, e à limitação intrínseca do algoritmo, que busca fazer um mapeamento linear entre perturbações no domínio e leituras nos fotodiodos, que é uma primeira aproximação da realidade.

# 7. INTEGRAÇÃO ENTRE *HARDWARE* E *SOFTWARE*: OBTENÇÃO DE IMAGENS TOMOGRÁFICAS

Até o presente momento, a construção e os testes do *hardware* e do *software* trilharam caminhos paralelos, porém separados. Nesse capítulo, descreve-se a integração entre ambos. O *hardware* será empregado na obtenção de medidas, as quais serão posteriormente processadas pelo *software*, com o objetivo de se obter imagens tomográficas.

Para se atingir esse fim, o *hardware* foi preparado conforme previsto inicialmente. A cuba de plástica foi preenchida com uma mistura de 500ml de água e 200ml de leite, de forma a se obter uma mistura bastante turva, como mostra a Ilustração 37:

**Ilustração 37: Cuba preparada para o teste final.**

Todos os LED's e fotodiodos foram completamente imersos na mistura. Adicionalmente, o limite de espessura de 3cm no qual ocorre difusão foi respeitado (isto é, existe mais que 1,5cm entre a superfície da mistura e a matriz de LED's e fotodiodos).

Com a mistura preparada, o protótipo foi tampado e isolado opticamente, conforme explicado anteriormente. Então, duas seqüências de medidas foram realizadas. A primeira delas, com o protótipo tal qual está mostrado acima: sem corpos estranhos, apenas com a mistura. A segunda seqüência foi feita com uma bolinha preta e opaca de 5cm de diâmetro imersa na mistura. Sua posição estava ligeiramente deslocada em relação ao centro da cuba.

Os resultados das duas seqüências de medidas encontram-se resumidos na Tabela 11. Assim como nos testes de *hardware* (vide capítulo 4), cada medida foi feita 20 vezes. Os desvios desses valores foram desprezíveis, de forma que aqui só se registraram os valores médios. Por perturbação normalizada, entenda-se a variação relativa da medida perturbada em relação à não perturbada.

**Tabela 11: Medidas tomadas no teste final.**

| LED aceso | Fotodiodo de leitura | Medida não perturbada | Medida perturbada | Perturbação normalizada |
|---|---|---|---|---|
| 1 | 2 | 0,1370 | 0,1517 | 1,11 |
| | 3 | 0,2099 | 0,2346 | 1,12 |
| | 4 | 0,0577 | 0,0644 | 1,11 |
| | 5 | 0,2743 | 0,3032 | 1,11 |
| | 6 | 0,2271 | 0,2568 | 1,13 |
| 2 | 1 | 0,1422 | 0,1582 | 1,11 |
| | 3 | 0,2099 | 0,2146 | 1,02 |
| | 4 | 0,0586 | 0,0631 | 1,08 |
| | 5 | 0,2765 | 0,2898 | 1,05 |
| | 6 | 0,2279 | 0,2345 | 1,03 |
| 3 | 1 | 0,1448 | 0,1613 | 1,11 |
| | 2 | 0,1363 | 0,1412 | 1,04 |
| | 4 | 0,0611 | 0,0757 | 1,24 |
| | 5 | 0,2783 | 0,2720 | 0,98 |
| | 6 | 0,2303 | 0,2465 | 1,07 |
| 4 | 1 | 0,1451 | 0,1606 | 1,11 |
| | 2 | 0,1344 | 0,1412 | 1,05 |
| | 3 | 0,2088 | 0,2594 | 1,24 |
| | 5 | 0,2760 | 0,3312 | 1,20 |
| | 6 | 0,2277 | 0,2486 | 1,09 |
| 5 | 1 | 0,1447 | 0,1595 | 1,10 |
| | 2 | 0,1348 | 0,1403 | 1,04 |
| | 3 | 0,2062 | 0,2006 | 0,97 |
| | 4 | 0,0649 | 0,0796 | 1,23 |
| | 6 | 0,2291 | 0,2320 | 1,01 |
| 6 | 1 | 0,1470 | 0,1679 | 1,14 |
| | 2 | 0,1352 | 0,1410 | 1,04 |
| | 3 | 0,2044 | 0,2195 | 1,07 |
| | 4 | 0,0584 | 0,0645 | 1,11 |
| | 5 | 0,2778 | 0,2842 | 1,02 |

A normalização da medida é necessária porque os fotodiodos não captam a densidade de fótons propriamente dita, e sim uma função presumidamente linear da mesma. Logo, com a perturbação normalizada em mãos, é possível calcular o vetor $\Lambda$, das perturbações de leitura nos fotodiodos devido à aparição de perturbações de absorção no domínio. Para tanto, um tratamento de dados adicional é necessário.

A matriz $\Lambda_0$ já foi calculada pelo programa (vide capítulo 5.5). O valor numérico de seus elementos está registrado na Tabela 12. Com a perturbação normalizada, pode-se calcular o vetor $\Lambda$, que é equivalente à diferença entre o caso perturbado e o não perturbado.

**Tabela 12: Processamento de dados anterior à execução do programa de problema inverso.**

| LED aceso | Fotodiodo de leitura | $\Lambda_0$ | Perturbação normalizada | $\Lambda$ |
|---|---|---|---|---|
| 1 | 2 | 214 | 1,11 | 23,10 |
| | 3 | -364 | 1,12 | -42,86 |
| | 4 | -614 | 1,11 | -70,33 |
| | 5 | -359 | 1,11 | -37,82 |
| | 6 | 220 | 1,13 | 28,70 |
| 2 | 1 | 214 | 1,11 | 24,07 |
| | 3 | 202 | 1,02 | 4,53 |
| | 4 | -360 | 1,08 | -27,82 |
| | 5 | -604 | 1,05 | -29,10 |
| | 6 | -358 | 1,03 | -10,36 |
| 3 | 1 | -364 | 1,11 | -41,43 |
| | 2 | 202 | 1,04 | 7,30 |
| | 4 | 200 | 1,24 | 47,48 |
| | 5 | -367 | 0,98 | 8,28 |
| | 6 | -619 | 1,07 | -43,53 |
| 4 | 1 | -614 | 1,11 | -65,62 |
| | 2 | -360 | 1,05 | -18,09 |
| | 3 | 200 | 1,24 | 48,46 |
| | 5 | 197 | 1,20 | 39,38 |
| | 6 | -364 | 1,09 | -33,43 |
| 5 | 1 | -359 | 1,10 | -36,52 |
| | 2 | -604 | 1,04 | -24,70 |
| | 3 | -367 | 0,97 | 9,98 |
| | 4 | 197 | 1,23 | 44,63 |
| | 6 | 212 | 1,01 | 2,72 |
| 6 | 1 | 220 | 1,14 | 31,31 |
| | 2 | -358 | 1,04 | -15,20 |
| | 3 | -619 | 1,07 | -45,81 |
| | 4 | -364 | 1,11 | -38,48 |
| | 5 | 212 | 1,02 | 4,93 |

Finalmente, esse vetor $\Lambda$ foi multiplicado (à direita) pela matriz $B$ (previamente calculada pelo programa), de forma a se obter o vetor $\Theta$, das perturbações de absorção no domínio. Com esse vetor e o valor não perturbado de absorção, pôde-se calcular a distribuição de absorção no domínio, a qual foi desenhada em gráfico, conforme mostra a Ilustração 38:

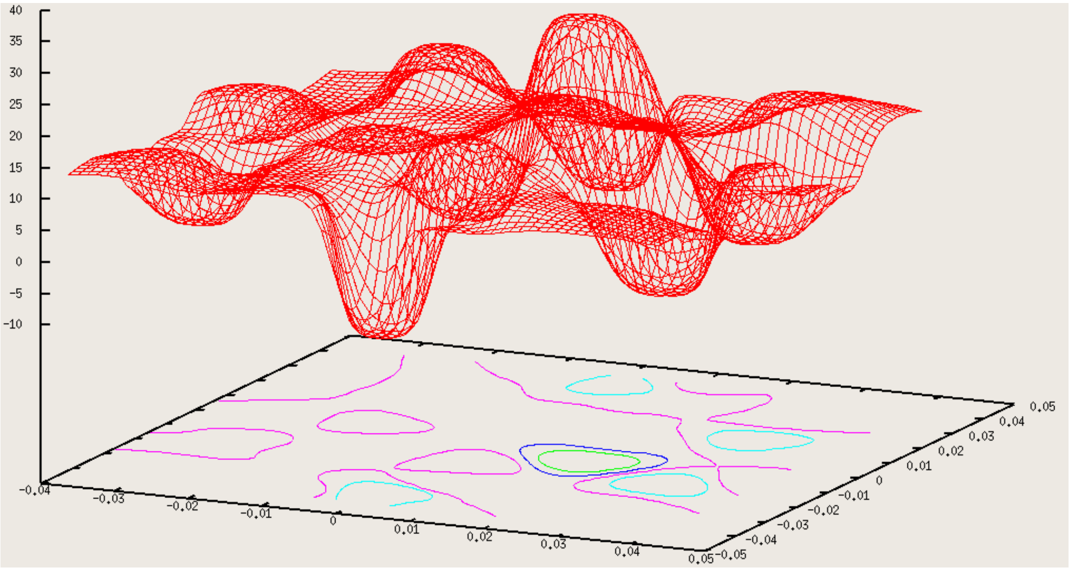



**Ilustração 38: Imagem tomográfica do teste final (3D).**

Como se pode verificar, as oscilações são bastante expressivas em todo o domínio. No entanto, existe uma região com absorção particularmente alta, próximo ao centro da figura. Essa região coincide com a posição da bola preta e opaca.

A Ilustração 39 mostra mais claramente a posição e a dimensão da anomalia:

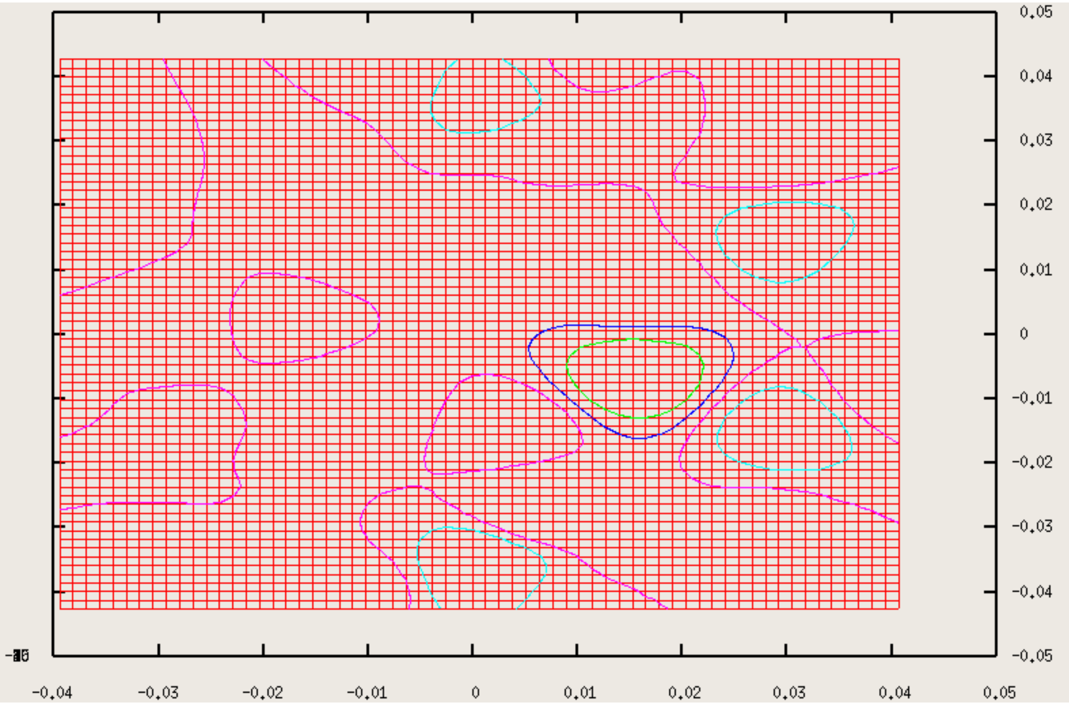



**Ilustração 39: Imagem tomográfica do teste final (2D).**

Como se verifica, a posição da anomalia (indicada pelos contornos verde e azul) está correta. Seu diâmetro é de aproximadamente 2,5cm, que é a metade do diâmetro da bolinha.

# 8. DISCUSSÃO E CONCLUSÕES

A imagem obtida no teste final realmente identificou uma perturbação na distribuição de absorção no domínio considerado. Sua posição coincide com a posição do objeto preto e opaco colocado na cuba do protótipo. Adicionalmente, a imagem mostra que ela é aproximadamente circular (como era esperado), e seu diâmetro é aproximadamente a metade do esperado. Isso comprova a eficácia do protótipo e do programa para determinar a existência, posição e, dentro de certa aproximação, a dimensão do objeto estranho no domínio.

Por outro lado, a qualidade da imagem não é boa, por dois motivos principais. Em primeiro lugar, porque o valor do coeficiente de absorção é negativo em alguns pontos do domínio. Em segundo lugar, porque as oscilações desse valor são grandes até mesmo em regiões onde deveria ser constante. Essa baixa qualidade deve-se à regularização empregada, que é pobre, ao refinamento da malha, que é grosseiro, e ao número de sensores, que é baixo.

O primeiro objetivo do Trabalho foi atingido: construção do protótipo do tomógrafo óptico de primeira geração. O segundo objetivo, que consiste na confecção do programa que permite empregá-lo para a obtenção de imagens tomográficas, também foi atingido. Com o *hardware* e o *software* em mãos, foi possível realizar o teste final, que consiste na determinação de uma imagem real a partir de medidas reais.

Como próximos passos na empreitada que este projeto começou, recomendam-se algumas sugestões de melhoria, a ver. O protótipo atual é muito sensível a mudanças de concentração da mistura. Isto é, as medidas variam de forma muito brusca em função da concentração de leite na água. Por isso, recomenda-se a adoção de uma mistura padrão em todas as experiências. Em segundo lugar, as medidas variam em torno de 50% entre um par LED-foto-diodo e outro. Logo, é recomendável que um protótipo de próxima geração seja construído com resistores, sensores e receptores de luz previamente testados e selecionados de forma a ser o mais uniforme possível. Em terceiro lugar, o fato de se adotar raios infravermelhos comprometeu a compreensão física do fenômeno de difusão da luz na mistura. Recomenda-se, portanto, que um próximo protótipo seja construído com LED's que emitam luz visível, de forma que o pesquisador possa, literalmente, enxergar o que está acontecendo na cuba.

Quanto ao programa, algumas sugestões também podem ser feitas. Ele foi construído como um problema bidimensional de difusão de luz. No entanto, testes empregando um LED com luz visível indicam que a difusão de luz ocorre de forma considerável no eixo z. Logo, recomenda-se que o *software* de segunda geração seja tridimensional por natureza. Em segundo lugar, o programa se apóia sobre a equação de Helmholtz para descrever o transporte de fótons em meio altamente difusivo. No entanto, essa é uma aproximação que limita a aplicabilidade do modelo a situações na qual a difusão é muito intensa. Por isso, recomenda-se pesquisa bibliográfica adicional no sentido de comprovar a validade de equação de Helmholtz em uma gama mais extensa de aplicações, ou no sentido de se adotar outra equação física que descreva o problema de forma mais abrangente.

## ANEXO A CÓDIGO-FONTE (EM C) DO PROGRAMA DE CONTROLE DA PLACA DE AQUISIÇÃO DE DADOS

```
#include <stdio.h>
#include <math.h>
#include "nrutil.h"
#include "nr.h"
#include <sys/io.h>
#include <time.h>

/* nmax = # de leituras */
#define nmax 20

/* nchannel = canais a
serem lidos
 * comeca no zero
 */
#define ncanais 6

#define ndelay 44000

#define
CLOCK_PER_SEC 1000000.0

/* Variavel global que vai
guardar os valores lidos */
double
canal[ncanais][nmax];

void lecanal(int,int);

int main(void) {
 int naux,i,j,x;
 double delta=0.0005;
 double time1,time2,dtime;
 FILE *fp;
```

```
/* Pede permissao para escrever em
portas */
naux=iopl(3); /* Set I/O permission
*/
printf("Permission enabled
iopl=%d\n",naux);

/* Inicia a placa PCL 711 */
/* Inicia programacao de modo */
/* BASE=512. Logo: 520 = BASE
+ 8 */
outb(0,520);  /* Clear interrupt flag
*/
outb(0,521);  /* 000 = ganho x1
(5V), 001=x2 (2,5V) */
        /* 010=x4 (1,25V),
011=x8 (0,625V) */
        /* 100=x16 (0,3125V) */
outb(0,522);  /* Multiplexer: canal
de leitura 0 */
outb(0,523);  /* Mode and interrupt
control register */
        /* 0 = Software transfer
and trigger */

time1=clock()/CLOCK_PER_SEC;

printf("\nDigite o ganho
(0,1,2,3,4):");
scanf("%d",&x); /* Definicao do
ganho de 2^x vezes */
```

```
fp=fopen("out.txt","w"); /* Abre arquivo para gravacao */
/* Loop de leitura com LEDs desativados, e depois 1 ativado por vez */
for (naux=0;naux<=6;naux++) {
 switch (naux) {
  case 0:outb(0,525);outb(0,526);fprintf(fp,"\nLEDs desligados\n");break;
   /* Desativa LEDs */
  case 1:outb(1,525);outb(0,526);fprintf(fp,"\nLED 1 ativado\n");break;
   /* Ativa so LED 1 */
  case 2:outb(2,525);outb(0,526);fprintf(fp,"\nLED 2 ativado\n");break;
   /* Ativa so LED 2 */
  case 3:outb(4,525);outb(0,526);fprintf(fp,"\nLED 3 ativado\n");break;
   /* Ativa so LED 3 */
  case 4:outb(8,525);outb(0,526);fprintf(fp,"\nLED 4 ativado\n");break;
   /* Ativa so LED 4 */
  case 5:outb(16,525);outb(0,526);fprintf(fp,"\nLED 5 ativado\n");break;
   /* Ativa so LED 5 */
  case 6:outb(32,525);outb(0,526);fprintf(fp,"\nLED 6 ativado\n");break;
   /* Ativa so LED 6 */
  }
 for (i=0;i<ndelay;i++) {
  for (j=0;j<ndelay;j++); /* Delay antes de amostrar */
 }
  /* Amostragem dos canais 0-5 */
  for (i=0;i<=5;i++) {
   lecanal(i,x);
  }
  /* Gravacao das leituras no arquivo */
  for (j=0;j<=nmax-1;j++) {
   fprintf(fp,"%d %f %f %f %f %f %f\n",j,canal[0][j],canal[1][j],canal[2][j],canal[3][j],canal[4][j],canal[5][j]);
  }
 }
 fclose(fp); /* Fecha o arquivo de gravacao */
 outb(0,525);outb(0,526); /* Desliga todos os LEDs */
 /* Marca o instante em que terminou a sequencia de leituras */
 time2=clock()/CLOCK_PER_SEC;
   dtime=time2-time1;
   printf("dtime %f\n",dtime);
   printf("time1 %f\n",time1);
   printf("time2 %f\n",time2);
 /* Desabilita acesso as portas */
```

```
naux=iopl(0); /* Set I/O
permission */
printf("Portas desabilitadas
iopl=%d\n",naux);
}

void lecanal(int k, int x) {
/* Objetivo: ler o canal k
da placa de aquisicao,
 * com um ganho de x
vezes, e nmax amostragens
 */
int aux,high,low,j;
double volt,escala;

for (j=0;j<=nmax-1;j++) {
 outb(k,522); /*
Multiplex: canal k */
 outb(x,521); /* Ganho de
2^x vezes */
 outb(1,524); /* Dispara
SW-trigger */

 /* Delay entre trigger e
amostragem */
 for
(aux=1;aux<ndelay;aux++);

 /* Verifica se conversao
A/D terminou */
 high=inb(517); /* Leitura
do A/D high byte */
 while (high>=16) { /*
high>=16 -> D4=1 ->
 * DRDY=1 ->
conversao
 * A/D incompleta
 */
  high=inb(517);
 }

 /* Conversao OK: posso ler o
byte inferior */
 low=inb(516);

 /* Fundo de escala, dado o ganho
*/
 switch (x) {
  case 0:escala=5.0;break;
  case 1:escala=2.5;break;
  case 2:escala=1.25;break;
  case 3:escala=0.625;break;
  case 4:escala=0.3125;break;
 }
 /* Conversao para volts */

volt=((high*256.0+low*1.0)/2048.0-
1.0)*escala;
 /* Guarda o valor no vetor
correspondente */
 canal[k][j]=volt;
}
}
```

## ANEXO B CÓDIGO-FONTE (EM C) DO PROGRAMA DE ELEMENTOS FINITOS RELATIVO À DIFUSÃO DE LUZ

**Logan.h (arquivo de *header*)**

```
/* This header file contains the
declaration of all functions implemented
on the c files */
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>


void calcula_Y(gsl_vector *D,
          gsl_matrix *nlk,
          gsl_matrix_int *nodes,
          gsl_matrix *Yinv,
          gsl_vector *mu,
          gsl_matrix *hij);

void init_centroids(gsl_vector *xc,
            gsl_vector *yc,
            gsl_vector *xg,
            gsl_vector *yg,
            gsl_matrix_int *nodes);

void init_cd(gsl_vector *xc,
         gsl_vector *yc,
         gsl_vector_int *numbering);

void init_nodes(gsl_matrix_int *nodes);

void init_matrices(gsl_vector *xc,
            gsl_vector *yc,
            gsl_vector *xg,
            gsl_vector *yg,
            gsl_matrix_int *nodes,
            gsl_matrix *nkl,
            gsl_matrix *hij,
            double thickness);

void init_potenciais(gsl_vector_int
*potenciais_conhecidos,
                gsl_vector *potenciais);

void init_vazao(gsl_vector_int
*n_vazao_conhecida,
               gsl_vector *fonte);

void generate_Y(gsl_vector *D,
            gsl_matrix *nlk,
            gsl_vector *mu,
            gsl_matrix *hij,
            gsl_matrix *Y,
            gsl_matrix_int *nodes );

void renumber(gsl_vector_int
*numbering,
          gsl_matrix_int *topology);

void generate_phi(gsl_matrix *Yinv,
            gsl_vector_int *leds,
            gsl_vector *phi_big,
            double ledpwr);
```

**Teste.c (função *main*)**

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>
#include <gsl/gsl_linalg.h>
#include <gsl/gsl_blas.h>
#include "logan.h"

#define ne 32
/* number of finite elements */
#define nnp 23
/* number of nodes */
#define npairs 6
/* number of LED/Photo-diode pairs */
#define n_potenciais 0
/* number of nodes whose potential is
known */
#define n_vazao 1
/* number of nodes whose flow is known
*/
#define nnp_local 3
/* number of nodes of the finite element
(3=triangle) */
#define thickness 0.03
/* height */
```

```
#define mu_0 10.0
/* absorption coefficient (at first) */
/* according to Guven et alli, 2003:
mu_0=5m^-1 */
#define ms 30.0
/* reduced scattering coefficient */
/* according to Guven et alli, 2003:
ms=1000m^-1 */
#define chg 0.5
/* percentual change in mu  */
#define alfa 0.00001
/* Blackbox algorithm matrix inversion
correction factor */

int  main() {
/* Memory allocation for scalars */
int i,j,k,s;
double aux,ledpwr;

/* Memory allocation for vectors and
matrices (GSL library) */
gsl_vector *x=gsl_vector_alloc(ne);
gsl_vector *xg=gsl_vector_alloc(ne);
gsl_vector *yg=gsl_vector_alloc(ne);
gsl_vector *xc=gsl_vector_alloc(nnp);
gsl_vector *yc=gsl_vector_alloc(nnp);
gsl_vector_int
*potenciais_conhecidos=gsl_vector_int_a
lloc(n_potenciais+1);
gsl_vector
*potenciais=gsl_vector_alloc(n_potenciai
s+1);
gsl_vector_int
*n_vazao_conhecida=gsl_vector_int_allo
c(n_vazao+1);
gsl_matrix
*Yinv=gsl_matrix_alloc(nnp,nnp);
gsl_matrix_int
*nodes=gsl_matrix_int_alloc(ne,nnp_loc
al);
gsl_matrix
*nlk=gsl_matrix_alloc(ne,nnp_local*nnp
_local);
gsl_matrix
*hij=gsl_matrix_alloc(ne,nnp_local*nnp
_local);
gsl_vector *mu=gsl_vector_alloc(ne);
gsl_vector *D=gsl_vector_alloc(ne);
gsl_vector *fonte=gsl_vector_alloc(nnp);
gsl_vector *phi=gsl_vector_alloc(nnp);
gsl_vector_int
*numbering=gsl_vector_int_alloc(nnp);
gsl_vector_int
*leds=gsl_vector_int_alloc(npairs+1);
gsl_matrix
*teta=gsl_matrix_alloc(ne,ne);
gsl_matrix
*B=gsl_matrix_alloc(ne,npairs*(npairs-
1));
gsl_vector
*phi_big_0=gsl_vector_alloc(npairs*(npa
irs-1));
gsl_vector
*phi_big=gsl_vector_alloc(npairs*(npairs
-1));
gsl_matrix
*lambda=gsl_matrix_alloc(npairs*(npairs
-1),ne);
gsl_matrix
*LLT=gsl_matrix_alloc(npairs*(npairs-
1),npairs*(npairs-1));
gsl_permutation
*p=gsl_permutation_alloc(npairs*(npairs
-1));
gsl_matrix
*LLTinv=gsl_matrix_alloc(npairs*(npair
s-1),npairs*(npairs-1));
gsl_matrix
*TLT=gsl_matrix_alloc(ne,npairs*(npair
s-1));

/* Memory allocation for files */
FILE *fp;

/* HERE'S WHERE THE FORWARD
PROBLEM SOLVING BEGINS */

/* Read nodal coordinates from
"coords.dat" file, and record
 * on matrices "x_c" and "y_c" */
init_cd(xc,yc,numbering);

/* Read topology from "nodes.dat", and
record on matrix "nodes" */
init_nodes(nodes);
```

```
renumber(numbering,nodes);
/* Define centroids for each element, and
record on matrices
 * "x_g" and "y_g" */
init_centroids(xc,yc,xg,yg,nodes);

/* Generate a tensor "nlk" with all local
matrices, i.e.
 * [B]T*[I]*[B]*t/4A, and another tensor
"hij" with all
 * [N]t*[N]*t/48A matrices */
init_matrices(xc,yc,xg,yg,nodes,nlk,hij,th
ickness);

/* Now, the undisturbed measurements
are determined */

/* Define the absorption vector "mu"
(initial) */
gsl_vector_set_all(mu,mu_0);

/* Define the diffusion vector "D"
(initial) */
for(i=0;i<ne;i++) {
  aux=gsl_vector_get(mu,i);
  gsl_vector_set(D,i,1/(3*(aux+ms)));
}
calcula_Y(D,nlk,nodes,Yinv,mu,hij);

/* Reads the value of photon flux
generated by a LED, and
 * redords it on "ledpwr". Afterwards,
reads the number
 * of the nodes in which a LED / photo-
diode exists, and
 * records them onto the vector "leds" */
fp=fopen("leds.dat","r");
fscanf(fp,"%lg",&ledpwr);
for(k=0;k<npairs;k++){
  fscanf(fp,"%d",&i);
gsl_vector_int_set(leds,k,i);
}
fclose(fp);

/* Calculates the undisturbed
measurements */
generate_phi(Yinv,leds,phi_big,ledpwr);
gsl_vector_memcpy(phi_big_0,phi_big);

/* Draw four examples of the forward
problem */
/* First, the undisturbed condition with 1
LED */
gsl_vector_set_all(fonte,0);
gsl_vector_set(fonte,0,ledpwr);
gsl_blas_dgemv(CblasNoTrans,1.0,Yinv,
fonte,0.0,phi);
fp=fopen("graf3d.dat","w");
for(i=0;i<nnp;i++){
 fprintf(fp,"%lg %lg
%lg\n",gsl_vector_get(xc,i),gsl_vector_g
et(yc,i),gsl_vector_get(phi,i));
}
fclose(fp);
system("gnuplot grafico.gnu");

/* Second, the undisturbed condition with
all the LEDs */
gsl_vector_set_all(fonte,0);
gsl_vector_set(fonte,0,ledpwr);
gsl_vector_set(fonte,10,ledpwr);
gsl_vector_set(fonte,9,ledpwr);
gsl_vector_set(fonte,2,ledpwr);
gsl_vector_set(fonte,6,ledpwr);
gsl_vector_set(fonte,5,ledpwr);
gsl_blas_dgemv(CblasNoTrans,1.0,Yinv,
fonte,0.0,phi);
fp=fopen("graf3d.dat","w");
for(i=0;i<nnp;i++){
 fprintf(fp,"%lg %lg
%lg\n",gsl_vector_get(xc,i),gsl_vector_g
et(yc,i),gsl_vector_get(phi,i));
}
fclose(fp);
system("gnuplot grafico.gnu");

/* Third, a disturbed condition (greater
mu) with 1 LED */
gsl_vector_set_all(fonte,0);
gsl_vector_set(fonte,0,ledpwr);
aux=mu_0*(1.0+chg);

gsl_vector_set(mu,19,aux);
gsl_vector_set(D,19,1/(3*(aux+ms)));
gsl_vector_set(mu,18,aux);
gsl_vector_set(D,18,1/(3*(aux+ms)));
```

```
gsl_vector_set(mu,5,aux);
gsl_vector_set(D,5,1/(3*(aux+ms)));
gsl_vector_set(mu,10,aux);
gsl_vector_set(D,10,1/(3*(aux+ms)));


calcula_Y(D,nlk,nodes,Yinv,mu,hij);
gsl_blas_dgemv(CblasNoTrans,1.0,Yinv,
fonte,0.0,phi);
fp=fopen("graf3d.dat","w");
for(i=0;i<nnp;i++){
 fprintf(fp,"%lg %lg
%lg\n",gsl_vector_get(xc,i),gsl_vector_g
et(yc,i),gsl_vector_get(phi,i));
}
fclose(fp);
system("gnuplot grafico.gnu");

/* Forth, a disturbed condition (zero mu)
with 1 LED */
gsl_vector_set_all(fonte,0);
gsl_vector_set(fonte,0,ledpwr);
aux=mu_0*0.0;

gsl_vector_set(mu,19,aux);
gsl_vector_set(D,19,1/(3*(aux+ms)));
gsl_vector_set(mu,18,aux);
gsl_vector_set(D,18,1/(3*(aux+ms)));
gsl_vector_set(mu,5,aux);
gsl_vector_set(D,5,1/(3*(aux+ms)));
gsl_vector_set(mu,10,aux);
gsl_vector_set(D,10,1/(3*(aux+ms)));
gsl_vector_set(mu,28,aux);
gsl_vector_set(D,28,1/(3*(aux+ms)));
gsl_vector_set(mu,4,aux);
gsl_vector_set(D,4,1/(3*(aux+ms)));
gsl_vector_set(mu,21,aux);
gsl_vector_set(D,21,1/(3*(aux+ms)));
gsl_vector_set(mu,27,aux);
gsl_vector_set(D,27,1/(3*(aux+ms)));
gsl_vector_set(mu,22,aux);
gsl_vector_set(D,22,1/(3*(aux+ms)));


calcula_Y(D,nlk,nodes,Yinv,mu,hij);
gsl_blas_dgemv(CblasNoTrans,1.0,Yinv,
fonte,0.0,phi);
fp=fopen("graf3d.dat","w");
for(i=0;i<nnp;i++){
 fprintf(fp,"%lg %lg
%lg\n",gsl_vector_get(xc,i),gsl_vector_g
et(yc,i),gsl_vector_get(phi,i));
}
fclose(fp);
system("gnuplot grafico.gnu");

/* HERE'S WHERE THE FORWARD
PROBLEM SOLVING ENDS */

/* HERE'S WHERE THE BLACKBOX
ALGORITHM BEGINS */
gsl_matrix_set_zero(teta);
gsl_matrix_set_zero(lambda);

for(j=0;j<ne;j++){
  /* Defines the absorption vector "mu"
(disturbed) */
 gsl_vector_set_all(mu,mu_0);

 /* Includes disturbance */
 aux=mu_0*(1.0+chg);
 gsl_vector_set(mu,j,aux);
 /* Records disturbance */
 gsl_matrix_set(teta,j,j,aux-mu_0);
 /* Define the diffusion vector "D"
(disturbed) */
 for(i=0;i<ne;i++) {
  aux=gsl_vector_get(mu,i);
  gsl_vector_set(D,i,1/(3*(aux+ms)));
 }
 calcula_Y(D,nlk,nodes,Yinv,mu,hij);
 generate_phi(Yinv,leds,phi_big,ledpwr);
 for(i=0;i<npairs*(npairs-1);i++){
  aux=gsl_vector_get(phi_big,i)-
gsl_vector_get(phi_big_0,i);
  gsl_matrix_set(lambda,i,j,aux);
 }
}

/* Record the matrices teta and lambda in
separate files */
fp=fopen("teta.dat","w");
for(i=0;i<ne;i++){
 for(j=0;j<ne;j++){
  fprintf(fp,"%lg
",gsl_matrix_get(teta,i,j));
```

```
 }
fprintf(fp,"\n");
}
fclose(fp);

fp=fopen("lambda.dat","w");
for(i=0;i<npairs*(npairs-1);i++){
 for(j=0;j<ne;j++){
  fprintf(fp,"%lg
",gsl_matrix_get(lambda,i,j));
 }
fprintf(fp,"\n");
}
fclose(fp);

gsl_matrix_set_identity(LLT);
gsl_blas_dgemm(CblasNoTrans,CblasTr
ans,1.0,lambda,lambda,alfa,LLT);
gsl_linalg_LU_decomp(LLT,p,&s);
gsl_linalg_LU_invert(LLT,p,LLTinv);
gsl_blas_dgemm(CblasNoTrans,CblasTr
ans,1.0,teta,lambda,0.0,TLT);
gsl_blas_dgemm(CblasNoTrans,CblasNo
Trans,1.0,TLT,LLTinv,0.0,B);

/* Record the matrix B in a separate file
*/
fp=fopen("B.dat","w");
for(i=0;i<ne;i++){
 for(j=0;j<npairs*(npairs-1);j++){
  fprintf(fp,"%lg ",gsl_matrix_get(B,i,j));
 }
fprintf(fp,"\n");
}
fclose(fp);

/* HERE'S WHERE THE BLACKBOX
ALGORITHM ENDS */

/* HERE'S WHERE THE PHANTOM
TESTS BEGIN */

/* Defines the absorption vector "mu" (a
little disturbed) */
gsl_vector_set_all(mu,mu_0);
/* Includes one disturbance */
aux=mu_0*(1.0+chg);
gsl_vector_set(mu,9,aux);

/* Define the diffusion vector "D"
(disturbed) */
for(i=0;i<ne;i++) {
 aux=gsl_vector_get(mu,i);
 gsl_vector_set(D,i,1/(3*(aux+ms)));
}
/* Forward problem */
calcula_Y(D,nlk,nodes,Yinv,mu,hij);
/* Find shadows */
generate_phi(Yinv,leds,phi_big,ledpwr);

/* Calculate shadow disturbances */
gsl_vector_sub(phi_big,phi_big_0);

/* Use B to determine mu disturbance
distribution */
gsl_blas_dgemv(CblasNoTrans,1.0,B,phi
_big,0.0,mu);

/* Calculate mu distribution */
for(i=0;i<ne;i++){

gsl_vector_set(mu,i,gsl_vector_get(mu,i)
+mu_0);
}
/* Record mu in a separate file */
fp=fopen("mu1.dat","w");
for (i=0;i<ne;i++) {
    fprintf(fp,"%d
%lg\n",i,gsl_vector_get(mu,i));
}
fclose(fp);
/* Draw the mu distribution */
fp=fopen("graf3d.dat","w");
for(i=0;i<ne;i++){
  fprintf(fp,"%lg %lg
%lg\n",gsl_vector_get(xg,i),

gsl_vector_get(yg,i),gsl_vector_get(mu,i)
);
}
fclose(fp);
system("gnuplot grafico.gnu");


/* Defines the absorption vector "mu"
(more disturbed) */
```

```
 gsl_vector_set_all(mu,mu_0);

/* Includes several disturbances */
aux=mu_0*(1.0+chg);
gsl_vector_set(mu,9,aux);
gsl_vector_set(mu,2,aux);

/* Define the diffusion vector "D"
(disturbed) */
for(i=0;i<ne;i++) {
 aux=gsl_vector_get(mu,i);
 gsl_vector_set(D,i,1/(3*(aux+ms)));
}
/* Forward problem */
calcula_Y(D,nlk,nodes,Yinv,mu,hij);
/* Find shadows */
generate_phi(Yinv,leds,phi_big,ledpwr);
/* Calculate shadow disturbances */
gsl_vector_sub(phi_big,phi_big_0);
/* Use B to determine mu disturbance
distribution */
gsl_blas_dgemv(CblasNoTrans,1.0,B,phi
_big,0.0,mu);
/* Calculate mu distribution */
for(i=0;i<ne;i++){

gsl_vector_set(mu,i,gsl_vector_get(mu,i)
+mu_0);
}
/* Record mu in a separate file */
fp=fopen("mu2.dat","w");
for (i=0;i<ne;i++) {
    fprintf(fp,"%d
%lg\n",i,gsl_vector_get(mu,i));
}
fclose(fp);
/* Draw the mu distribution */
fp=fopen("graf3d.dat","w");
for(i=0;i<ne;i++){
  fprintf(fp,"%lg %lg
%lg\n",gsl_vector_get(xg,i),

gsl_vector_get(yg,i),gsl_vector_get(mu,i)
);
}
fclose(fp);
system("gnuplot grafico.gnu");
```

```
/* Defines the absorption vector "mu"
(very disturbed) */
 gsl_vector_set_all(mu,mu_0);

/* Includes several disturbances */
aux=mu_0*(1.0+chg);
gsl_vector_set(mu,17,aux);
gsl_vector_set(mu,11,aux);
gsl_vector_set(mu,6,aux);
gsl_vector_set(mu,9,aux);

/* Define the diffusion vector "D"
(disturbed) */
for(i=0;i<ne;i++) {
 aux=gsl_vector_get(mu,i);
 gsl_vector_set(D,i,1/(3*(aux+ms)));
}
/* Forward problem */
calcula_Y(D,nlk,nodes,Yinv,mu,hij);
/* Find shadows */
generate_phi(Yinv,leds,phi_big,ledpwr);
/* Calculate shadow disturbances */
gsl_vector_sub(phi_big,phi_big_0);
/* Use B to determine mu disturbance
distribution */
gsl_blas_dgemv(CblasNoTrans,1.0,B,phi
_big,0.0,mu);
/* Calculate mu distribution */
for(i=0;i<ne;i++){

gsl_vector_set(mu,i,gsl_vector_get(mu,i)
+mu_0);
}
/* Record mu in a separate file */
fp=fopen("mu3.dat","w");
for (i=0;i<ne;i++) {
  fprintf(fp,"%d
%lg\n",i,gsl_vector_get(mu,i));
}
fclose(fp);

/* Draw the mu distribution */
fp=fopen("graf3d.dat","w");
for(i=0;i<ne;i++){
  fprintf(fp,"%lg %lg
%lg\n",gsl_vector_get(xg,i),
```

```
gsl_vector_get(yg,i),gsl_vector_get(mu,i)
);
}
fclose(fp);
system("gnuplot grafico.gnu");

/* HERE'S WHERE THE PHANTOM
TESTS END */

/* HERE'S WHERE THE REAL TEST
BEGINS */

/* Reads shadows measured, minus
undisturbed shadows */
gsl_vector_set_zero(phi_big);
gsl_vector_set(phi_big,0,23.1);
gsl_vector_set(phi_big,1,-42.9);
gsl_vector_set(phi_big,2,-70.3);
 gsl_vector_set(phi_big,3,-37.8);
gsl_vector_set(phi_big,4,28.7);
gsl_vector_set(phi_big,5,24.1);
gsl_vector_set(phi_big,6,4.5);
gsl_vector_set(phi_big,7,-27.8);
gsl_vector_set(phi_big,8,-29.1);
gsl_vector_set(phi_big,9,-10.4);
gsl_vector_set(phi_big,10,-41.4);
gsl_vector_set(phi_big,11,7.3);
gsl_vector_set(phi_big,12,47.5);
gsl_vector_set(phi_big,13,8.3);
gsl_vector_set(phi_big,14,-43.5);
gsl_vector_set(phi_big,15,-65.6);
gsl_vector_set(phi_big,16,-18.1);
gsl_vector_set(phi_big,17,48.5);
gsl_vector_set(phi_big,18,39.4);
gsl_vector_set(phi_big,19,-33.4);
gsl_vector_set(phi_big,20,-36.5);
gsl_vector_set(phi_big,21,-24.7);
gsl_vector_set(phi_big,22,10.0);
gsl_vector_set(phi_big,23,44.6);
gsl_vector_set(phi_big,24,2.7);
gsl_vector_set(phi_big,25,31.3);
gsl_vector_set(phi_big,26,-15.2);
gsl_vector_set(phi_big,27,-45.8);
gsl_vector_set(phi_big,28,-38.5);
gsl_vector_set(phi_big,29,4.9);

/* Use B to determine mu disturbance
distribution */
gsl_blas_dgemv(CblasNoTrans,1.0,B,phi
_big,0.0,mu);

/* Calculate mu distribution */
for(i=0;i<ne;i++){

gsl_vector_set(mu,i,gsl_vector_get(mu,i)
+mu_0);
}
/* Record mu in a separate file */
fp=fopen("mu_r.dat","w");
for (i=0;i<ne;i++) {
    fprintf(fp,"%d
%lg\n",i,gsl_vector_get(mu,i));
}
fclose(fp);
/* Draw the mu distribution */
fp=fopen("graf3d.dat","w");
for(i=0;i<ne;i++){
  fprintf(fp,"%lg %lg
%lg\n",gsl_vector_get(xg,i),

gsl_vector_get(yg,i),gsl_vector_get(mu,i)
);
}
fclose(fp);
system("gnuplot grafico.gnu");

/* HERE'S WHERE THE REAL TEST
ENDS */

/* Free allocated vector and matrix
memory */
gsl_vector_free(x);
gsl_vector_free(xg);
gsl_vector_free(yg);
gsl_vector_free(xc);
gsl_vector_free(yc);
gsl_vector_int_free(potenciais_conhecido
s);
gsl_vector_free(potenciais);
gsl_vector_int_free(n_vazao_conhecida);
gsl_matrix_free(Yinv);
gsl_matrix_int_free(nodes);
gsl_matrix_free(nlk);
gsl_matrix_free(hij);
```

```
gsl_vector_free(mu);
gsl_vector_free(D);
gsl_vector_int_free(numbering);
gsl_vector_int_free(leds);
gsl_matrix_free(teta);
gsl_matrix_free(B);
gsl_vector_free(phi_big_0);
gsl_vector_free(phi_big);
gsl_matrix_free(lambda);
gsl_matrix_free(LLT);
gsl_permutation_free(p);
gsl_matrix_free(LLTinv);
gsl_matrix_free(TLT);
return 0;
}
```

**Init_cd.c**

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <gsl/gsl_vector.h>


void init_cd(gsl_vector *xc,
          gsl_vector *yc,
          gsl_vector_int *numbering){

  int i,naux1,naux2;
  int nnp;
  double aux1,aux2;
  FILE *fp;

  fp=fopen("coord.dat","r");


  nnp=xc->size;

  for (i=0;i<nnp;i++) {

    fscanf(fp,"%d %lg %lg
%d",&naux1,&aux1,&aux2,&naux2);
    gsl_vector_set(xc,i,aux1);
    gsl_vector_set(yc,i,aux2);

gsl_vector_int_set(numbering,i,naux1);
  }
  fclose(fp);

}
```

**Init_nodes.c**

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <gsl/gsl_matrix.h>


void init_nodes(gsl_matrix_int *nodes) {

  int
i,naux1,naux2,naux3,naux4,naux5,naux6,
naux7,naux8;
  int nemax;
  FILE *fp;

  fp=fopen("topo.dat","r");

  nemax=nodes->size1;

  for (i=0;i<nemax;i++) {

    fscanf(fp,"%d %d %d %d %d %d %d
%d",&naux1,&naux2,&naux3,&naux4,

&naux5,&naux6,&naux7,&naux8);

    gsl_matrix_int_set(nodes,i,0,naux6);
    gsl_matrix_int_set(nodes,i,1,naux7);
    gsl_matrix_int_set(nodes,i,2,naux8);
  }

  fclose(fp);

}
```

**Renumber.c**

```
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>

void renumber(gsl_vector_int
*numbering,
          gsl_matrix_int *topology){

  int nnodes;
```

```
  int nelements;
  int nnp_local;
  int i;
  int j;
  int k;
  int naux1;
  int naux2;

  nnodes=numbering->size;
  nelements=topology->size1;
  nnp_local=topology->size2;

  for (i=0;i<nnodes;i++){

naux1=gsl_vector_int_get(numbering,i);
    for (j=0;j<nelements;j++){
      for (k=0;k<nnp_local;k++){
            naux2=gsl_matrix_int_get(topolo
gy,j,k);
        if (naux2==naux1){

gsl_matrix_int_set(topology,j,k,i+1);
            }
      }
    }
  }
}
```

**Init_centroids.c**

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>

void init_centroids(gsl_vector *xc,
                    gsl_vector *yc,
                    gsl_vector *xg,
                    gsl_vector *yg,
                    gsl_matrix_int *nodes){

  int i,naux1,naux2;
  int j;
  int ne;
  int nnp;
  int nnp_local;
  int node;
  double aux1,aux2;

  ne=nodes->size1;
  nnp=xc->size;
  nnp_local=nodes->size2;

  for (i=0;i<ne;i++) {

    aux1=0.0;
    aux2=0.0;
    for (j=0;j<nnp_local;j++){
      node=gsl_matrix_int_get(nodes,i,j)-1;
      aux1+=gsl_vector_get(xc,node);
      aux2+=gsl_vector_get(yc,node);
    }
    gsl_vector_set(xg,i,aux1/nnp_local);
    gsl_vector_set(yg,i,aux2/nnp_local);
  }

}
```

**Init_matrices.c**

```
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>

void init_matrices(gsl_vector *xc,
                   gsl_vector *yc,
                   gsl_vector *xg,
                   gsl_vector *yg,
                   gsl_matrix_int *nodes,
                   gsl_matrix *nlk,
                   gsl_matrix *hij,
                   double thickness) {

  int nnp;
  int ne;
  int nnp_local;
  int node;

  int i,j,q;
  int nh1,nh2;
  double aux,sxx,syy,sxy;

nnp=xc->size;
ne=nodes->size1;
nnp_local=nodes->size2;
```

```
gsl_vector
*a=gsl_vector_alloc(nnp_local);
gsl_vector
*b=gsl_vector_alloc(nnp_local);
gsl_vector
*d=gsl_vector_alloc(nnp_local);
gsl_vector *x1=gsl_vector_alloc(ne);
gsl_vector *y1=gsl_vector_alloc(ne);
gsl_vector *x2=gsl_vector_alloc(ne);
gsl_vector *y2=gsl_vector_alloc(ne);
gsl_vector *x3=gsl_vector_alloc(ne);
gsl_vector *y3=gsl_vector_alloc(ne);
gsl_vector *area=gsl_vector_alloc(ne);
gsl_matrix
*lk=gsl_matrix_alloc(nnp_local,nnp_loca
l);

/* Record x and y coords of nodes 1, 2, 3
(in the x1, x2, x3,
 * y1, y2, y3 matrices), for every FE (one
column per FE), and
 * translate origin of coords to centroid */
  for (i=0; i <ne; i++) {
    node=gsl_matrix_int_get(nodes,i,0)-1;
    aux=gsl_vector_get(xc,node)-
gsl_vector_get(xg,i);
    gsl_vector_set(x1,i,aux);
    aux=gsl_vector_get(yc,node)-
gsl_vector_get(yg,i);
    gsl_vector_set(y1,i,aux);

    node=gsl_matrix_int_get(nodes,i,1)-1;
    aux=gsl_vector_get(xc,node)-
gsl_vector_get(xg,i);
    gsl_vector_set(x2,i,aux);
    aux=gsl_vector_get(yc,node)-
gsl_vector_get(yg,i);
    gsl_vector_set(y2,i,aux);

    node=gsl_matrix_int_get(nodes,i,2)-1;
    aux=gsl_vector_get(xc,node)-
gsl_vector_get(xg,i);
    gsl_vector_set(x3,i,aux);
    aux=gsl_vector_get(yc,node)-
gsl_vector_get(yg,i);
    gsl_vector_set(y3,i,aux);
  }

/* Initialize the numerical local matrix
[B]t*[I]*[B]*t/4A,
 * and also the [N]t*[N]*t/48A matrix */
  gsl_matrix_set_zero(nlk);
  gsl_matrix_set_zero(hij);

/* For each FE... */
  for (q=0; q<ne; q++) {

/* ...calculate beta 1, 2, 3 */
    aux=gsl_vector_get(y2,q)-
gsl_vector_get(y3,q);
    gsl_vector_set(b,0,aux);
    aux=gsl_vector_get(y3,q)-
gsl_vector_get(y1,q);
    gsl_vector_set(b,1,aux);
    aux=gsl_vector_get(y1,q)-
gsl_vector_get(y2,q);
    gsl_vector_set(b,2,aux);

/* ...calculate gama 1, 2, 3 */
    aux=gsl_vector_get(x3,q)-
gsl_vector_get(x2,q);
    gsl_vector_set(d,0,aux);
    aux=gsl_vector_get(x1,q)-
gsl_vector_get(x3,q);
    gsl_vector_set(d,1,aux);
    aux=gsl_vector_get(x2,q)-
gsl_vector_get(x1,q);
    gsl_vector_set(d,2,aux);

/* ...calculate the FE's area and record it
in vector "area" */
    aux=(
gsl_vector_get(x2,q)*gsl_vector_get(y3,q
)+

gsl_vector_get(x3,q)*gsl_vector_get(y1,q
)+

gsl_vector_get(x1,q)*gsl_vector_get(y2,q
)-

gsl_vector_get(x2,q)*gsl_vector_get(y1,q
)-

gsl_vector_get(x3,q)*gsl_vector_get(y2,q
)-
```

```
gsl_vector_get(x1,q)*gsl_vector_get(y3,q
))/2;
    gsl_vector_set(area,q,aux);

/* ...calculate alfa 1, 2, 3 */

aux=gsl_vector_get(x2,q)*gsl_vector_get
(y3,q)-

gsl_vector_get(y2,q)*gsl_vector_get(x3,q
);
    gsl_vector_set(a,0,aux);

aux=gsl_vector_get(y1,q)*gsl_vector_get
(x3,q)-

gsl_vector_get(x1,q)*gsl_vector_get(y3,q
);
    gsl_vector_set(a,1,aux);

aux=gsl_vector_get(x1,q)*gsl_vector_get
(y2,q)-

gsl_vector_get(y1,q)*gsl_vector_get(x2,q
);
    gsl_vector_set(a,2,aux);

/* ...calculate sum(x^2), sum(y^2),
sum(x*y)
 * Note: origin of coords must be at the
centroid now */

sxx=(gsl_vector_get(x1,q)*gsl_vector_ge
t(x1,q))+

(gsl_vector_get(x2,q)*gsl_vector_get(x2,
q))+

(gsl_vector_get(x3,q)*gsl_vector_get(x3,
q));

syy=(gsl_vector_get(y1,q)*gsl_vector_ge
t(y1,q))+

(gsl_vector_get(y2,q)*gsl_vector_get(y2,
q))+

(gsl_vector_get(y3,q)*gsl_vector_get(y3,
q));

sxy=(gsl_vector_get(x1,q)*gsl_vector_ge
t(y1,q))+

(gsl_vector_get(x2,q)*gsl_vector_get(y2,
q))+

(gsl_vector_get(x3,q)*gsl_vector_get(y3,
q));

/* ...calculate [B]T*[I]*[B]*t/4A and
[N]t*[N]*t/48A */
    for (i=0; i<nnp_local; i++) {
      for (j=0; j<=i; j++) {
        aux=
thickness*(gsl_vector_get(b,i)*gsl_vecto
r_get(b,j)+

gsl_vector_get(d,i)*gsl_vector_get(d,j))/
              (4*gsl_vector_get(area,q));

gsl_matrix_set(nlk,q,j+i*nnp_local,aux);
        if (j!=i)
gsl_matrix_set(nlk,q,i+j*nnp_local,aux);

        aux=
thickness*(gsl_vector_get(a,i)*gsl_vector
_get(a,j)*12+

gsl_vector_get(b,i)*gsl_vector_get(b,j)*s
xx+

gsl_vector_get(d,i)*gsl_vector_get(d,j)*s
yy+

(gsl_vector_get(b,i)*gsl_vector_get(d,j)+

gsl_vector_get(b,j)*gsl_vector_get(d,i))*
sxy)/
              (48*gsl_vector_get(area,q));

gsl_matrix_set(hij,q,j+i*nnp_local,aux);
        if (j!=i)
gsl_matrix_set(hij,q,i+j*nnp_local,aux);
```

```
/* Note that: the [B]t*[I]*[B]*t/4A
matrix regarding the q-th
 * triangular FE is recorded in the q-th
line of matrix "nlk".
 * The lines have been put together side
by side.
 * Analogous explanation holds for
[N]t*[N]*t/48A and "hij" */
     }
    }
/* Repeat for the next FE */
  }

/* Free allocated vector and matrix
memory */
gsl_vector_free(b);
gsl_vector_free(d);
gsl_vector_free(x1);
gsl_vector_free(y1);
gsl_vector_free(x2);
gsl_vector_free(y2);
gsl_vector_free(x3);
gsl_vector_free(y3);
gsl_vector_free(area);
gsl_matrix_free(lk);
}
```

**Calcula_Y.c**

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h>
#include <math.h>
#include <malloc.h>
#include "logan.h"
#include <gsl/gsl_vector.h>
#include <gsl/gsl_linalg.h>
#include <gsl/gsl_matrix.h>

void calcula_Y(gsl_vector *D,
            gsl_matrix *nlk,
            gsl_matrix_int *nodes,
            gsl_matrix *Yinv,
            gsl_vector *mu,
            gsl_matrix *hij)
{
  int nnp,s;

 nnp=Yinv->size1;

gsl_permutation
*p=gsl_permutation_alloc(nnp);
gsl_matrix
*v=gsl_matrix_alloc(nnp,nnp);
gsl_matrix
*Y=gsl_matrix_alloc(nnp,nnp);
gsl_matrix
*H=gsl_matrix_alloc(nnp,nnp);

gsl_matrix_set_zero(Y);

generate_Y(D,nlk,mu,hij,Y,nodes);

gsl_matrix_memcpy(v,Y);

  /* Compute the inverse of the
permeability matrix  */
  gsl_linalg_LU_decomp(v,p,&s);
  gsl_linalg_LU_invert(v,p,Yinv);

gsl_permutation_free(p);
gsl_matrix_free(v);
gsl_matrix_free(Y);
}
```

**Generate_Y.c**

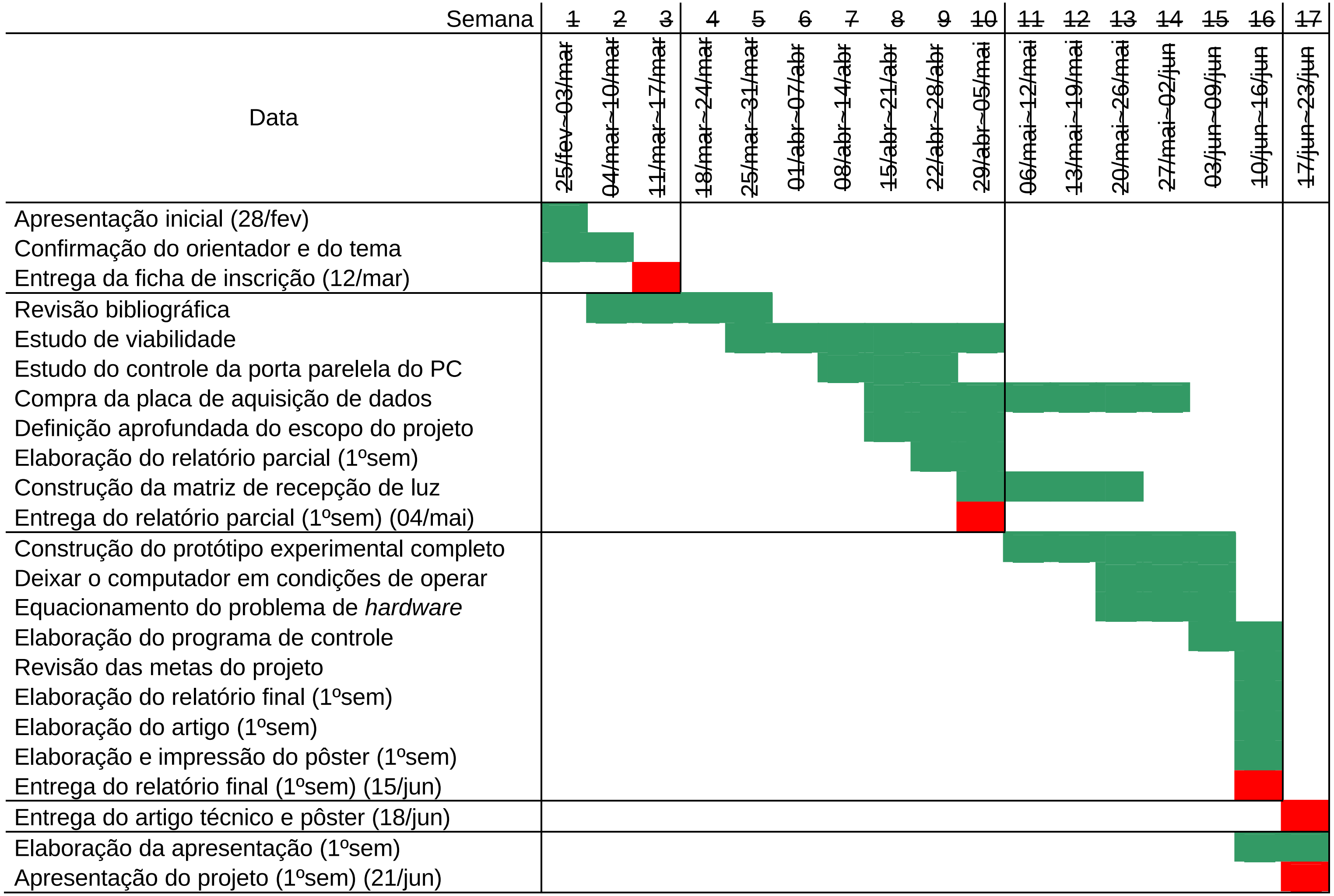
```
#include <stdio.h>
#include <gsl/gsl_vector.h>
#include <gsl/gsl_linalg.h>
#include <gsl/gsl_matrix.h>


void generate_Y(gsl_vector *D,
            gsl_matrix *nkl,
            gsl_vector *mu,
            gsl_matrix *hij,
            gsl_matrix *Y,
            gsl_matrix_int *nodes ){

int ne;
int nnp;
int nnp_local;
int i,j,q;
int nh1,nh2;
```

```
double aux1,aux2;

ne=D->size;
nnp=Y->size1;
nnp_local=nodes->size2;

/* For each FE... */
  for (q=0; q<ne; q++) {
/* ...for each line... */
    for (i=0; i<nnp_local; i++) {
/* ...for each column... */
      for (j=0; j<nnp_local; j++) {

aux1=(gsl_matrix_get(nkl,q,j+i*nnp_loca
l)*
           gsl_vector_get(D,q))-

(gsl_matrix_get(hij,q,j+i*nnp_local)*
           gsl_vector_get(mu,q));
        nh1=gsl_matrix_int_get(nodes,q,i)-
1;
        nh2=gsl_matrix_int_get(nodes,q,j)-
1;

aux2=gsl_matrix_get(Y,nh1,nh2)+aux1;
           gsl_matrix_set(Y,nh1,nh2,aux2);
          }
    }
  }
}
```

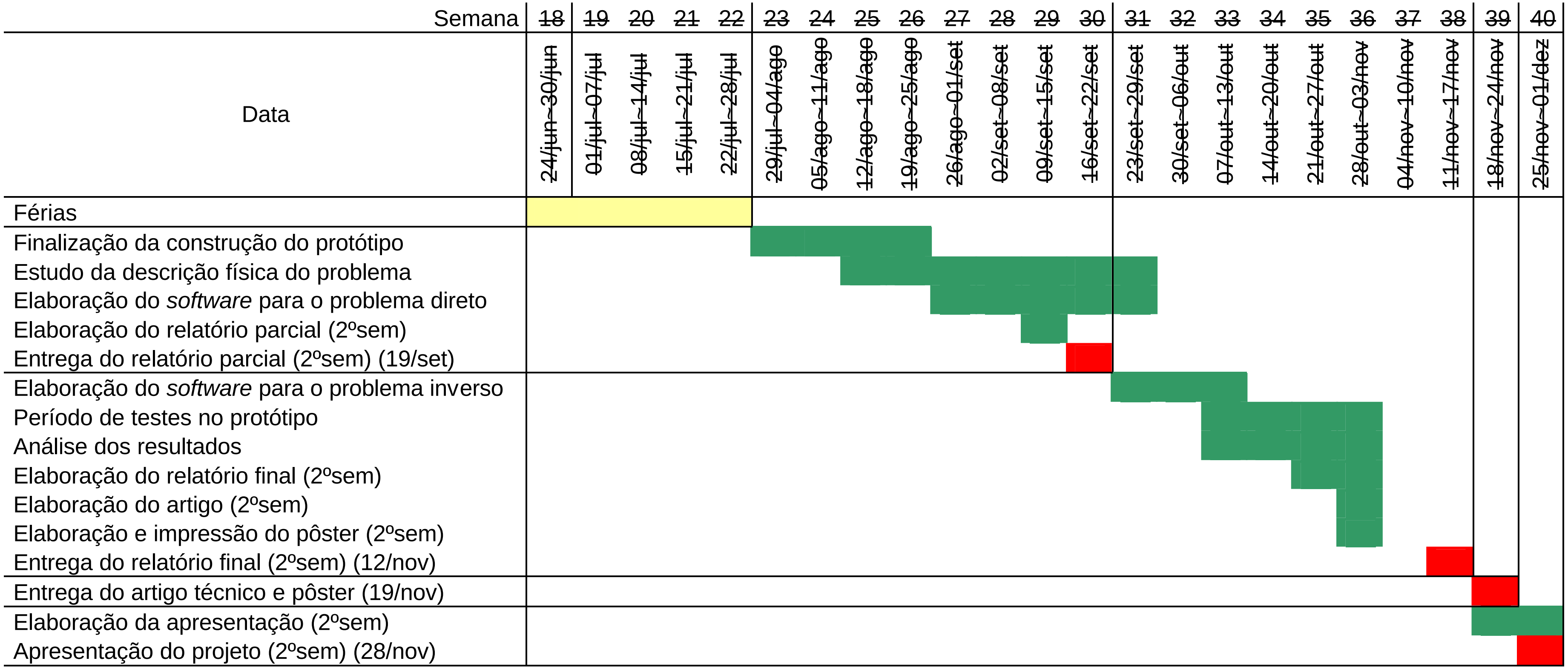
**Generate_phi.c**

```
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>
#include <gsl/gsl_blas.h>

void generate_phi(gsl_matrix *Yinv,
              gsl_vector_int *leds,
              gsl_vector *phi_big,
              double ledpwr){

int i,j,k,npairs,nnp;
npairs=leds->size;
npairs-=1;
nnp=Yinv->size1;
double aux;

gsl_vector *phi=gsl_vector_alloc(nnp);
gsl_vector *fonte=gsl_vector_alloc(nnp);

gsl_vector_set_zero(phi_big);

for(k=0;k<npairs;k++){
 i=gsl_vector_int_get(leds,k);
 gsl_vector_set_zero(fonte);
 gsl_vector_set(fonte,i-1,ledpwr);

gsl_blas_dgemv(CblasNoTrans,1.0,Yinv,
fonte,0.0,phi);
 for(i=0;i<npairs;i++){
  if(k!=i){
   j=gsl_vector_int_get(leds,i);
   aux=gsl_vector_get(phi,j-1);
   if(i>k)          gsl_vector_set(phi_big,i-
1+(npairs-1)*k,aux);
   if(i<k)
gsl_vector_set(phi_big,i+(npairs-
1)*k,aux);
  }
 }
}
gsl_vector_free(phi);
gsl_vector_free(fonte);
}
```

# ANEXO C CRONOGRAMA DO PROJETO – 1º SEMESTRE



## ANEXO D CRONOGRAMA DO PROJETO – 2º SEMESTRE




Legenda:
Atividades concluídas
Atividades marcadas pelo coordenador
Período sem atividades

# REFERÊNCIAS BIBLIOGRÁFICAS

[1] CAO, NANNAN & NEHORAI, ARYE. **Tumor localization using diffuse optical tomography and linearly constrained minimum variance beamforming.** Optical Society of America, 2007.

[2] ARRIDGE, S. R. **Optical tomography in medical imaging.** Inverse problems 15, R41-R93, 1999.

[3] GAUDETTE, R. J.; BROOKS, D. H.; DIMARZIO, C. A.; KILMER, M. E.; MILLER; E. L., GAUDETTE, T.; BOAS, D. A. **A comparison study of linear reconstruction techniques for diffuse optical tomographic imaging of absorption coefficient.** Phys. Med. Biol. 45, 1051–1070, 2000.

[4] VAN VEEN, B. D. & BUCKLEY, K. M. **Beamforming: a versatile approach to spatial filtering.** IEEE ASSP Magazine 5, 4–24, 1988.

[5] SYNNEVAG, J. F.; AUSTENG, A.; HOLM, S. **Minimum variance adaptive beamforming applied to medical ultrasound imaging.** Proceedings of IEEE Ultrasonics Symposium, IEEE, Rotterdam, Netherlands, pp. 1199–1202, 2005.

[6] MUELLER, J. **Examples of inverse problems: lecture 4.** Disponível em <http://www.math.colostate.edu/~mueller/inverse_problems/inverse_lecture4.pdf>.

[7] HOROWITZ, PAUL & HILL, WINFIELD. **The art of electronics.** Cambridge University Press, 1989.

[8] ZIENKIEWICZ, O. C. & TAYLOR, R. L. **The finite element method.** McGraw-Hill, 1989.

[9] LOGAN, Daryl L. **A first course in the finite element method using Algor™**, PWS publishing company, 1997.