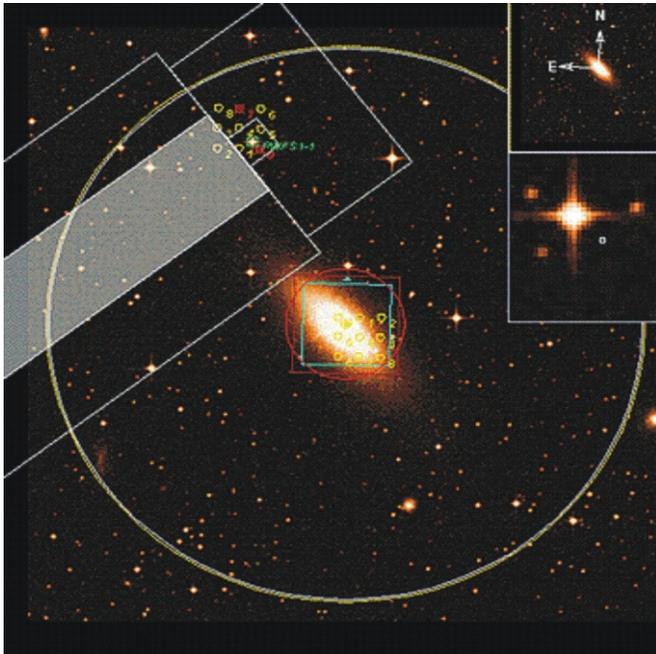


ISSN 1677-8464

## Implementação da Transformada de Fourier com o Java Advanced Imaging

José Iguelmar Miranda<sup>1</sup>  
João Camargo Neto<sup>2</sup>



O objetivo deste comunicado é apresentar a implementação Java™ da transformada discreta de Fourier e sua inversa, usando a biblioteca Java Advanced Imaging (JAI), para mapeamento entre os domínios espacial e de frequência em imagens digitais. A utilidade da transformada de Fourier, entre outras coisas, é que ela atua como um filtro, tanto de passa alta como de passa baixa. A implementação atende a necessidades de projetos desenvolvidos na Embrapa Informática Agropecuária na área de processamento de imagens aplicado a agropecuária.

As transformadas de Fourier foram desenvolvidas pelo matemático francês Jean Baptiste Joseph Fourier (1768-1830), quando estudando a teoria da propagação do calor (Fourier, 2003). Para ele, o calor, assim como a gravidade, penetra em toda substância do universo e sua causa primária está sujeita a leis constantes e simples. A partir desse raciocínio, ele desenvolveu as leis matemáticas da difusão do calor obedecidas pelos elementos. Uma de suas preocupações era a temperatura terrestre, tanto que lhe é conferida a paternidade do termo "efeito estufa", *effet de serre* (Prestini, 2003).

De uma maneira geral, "análise de Fourier" se refere ao uso de "séries" e "transformadas" de Fourier, sendo útil em vários campos da matemática, física, biologia etc. Para dar uma idéia inicial dessa análise, usemos, por exemplo, o cálculo vetorial; um vetor  $\mathbf{v}$  é uma entidade com as propriedades *comprimento* e *direção*. Normalmente, não se

realizam cálculos vetoriais diretamente com essas propriedades, mas usando as *componentes* do vetor. Por exemplo, no espaço bidimensional, elas seriam  $(v_1, v_2)$ . O comprimento de  $\mathbf{v}$  é calculado por:

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2}$$

Essas componentes são os coeficientes da representação única de  $\mathbf{v}$  como uma combinação linear de vetores na base padrão  $[i, j]$ :

$$\mathbf{v} = v_1\mathbf{i} + v_2\mathbf{j}$$

Como cada vetor pode ser representado dessa maneira e como  $[i, j]$  é uma base simples (seus elementos são ortogonais e de comprimento unitário), a manipulação de vetores pode ser reduzida a cálculos simples com as componentes de cada vetor. De maneira semelhante, quando Fourier estava tentando descrever matematicamente o processo de condução do calor, ele pensou em trabalhar com uma combinação linear de várias componentes. Para a condução do calor numa barra uniforme, ele representou a temperatura,  $f$ , na posição  $\theta$  da barra,  $f(\theta)$ , em um tempo determinado sujeito a certas condições iniciais e de contorno, como:

$$f(\theta) = a_0 + a_1 \cos(c\theta) + b_1 \sin(c\theta) + a_2 \cos(2c\theta) + b_2 \sin(2c\theta) + a_3 \cos(3c\theta) + b_3 \sin(3c\theta) + \dots \quad (1)$$

<sup>1</sup> Ph.D. em Geoprocessamento, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: miranda@cnptia.embrapa.br)

<sup>2</sup> Ph.D. em Processamento de Imagens, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: camargo@cnptia.embrapa.br)

onde  $c$  é  $\pi$  dividido pelo comprimento da barra, e os  $a_k$ 's e  $b_k$ 's são constantes a serem determinadas após condicionar essa representação para  $f$  nas equações modelando o fluxo de calor. A abordagem de Fourier foi correta<sup>3</sup>, e a sua idéia de representar uma função em termos de senos e cossenos levou ao desenvolvimento de várias aplicações na matemática (Howell, 2001).

## Séries e Transformadas de Fourier

Considere um fino anel circular. Para simplificar, considere que seu comprimento seja  $2\pi$ . Seja  $\theta$  a variável que designa a posição no anel (ângulo em radianos), de maneira que  $\theta$  e  $\theta + 2\pi$  fornecem a mesma posição no anel. A temperatura no tempo  $t$  e posição  $\theta$  é definida como  $f(t, \theta)$ . O fluxo de calor no anel é dada pela seguinte equação diferencial parcial (Jayakumar, 1998):

$$\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial \theta^2} \tag{2}$$

se a distribuição inicial de temperatura  $f(t, \theta) = f(0, \theta)$  é dada, esse problema é denominado problema de valor inicial para a equação do calor (2). Fourier propôs uma solução linear para o problema, expandindo a distribuição inicial de temperatura numa série trigonométrica.

Na palestra que Fourier apresentou na Academia Francesa, em dezembro de 1807, ele afirmou que *qualquer função arbitrária*, definida num intervalo finito, pode ser expressada como uma soma de funções de seno e cosseno. A validade do teorema foi gradualmente estendida e aceita com os trabalhos de Dirichlet em 1829 (Johann Peter Gustav Lejeune Dirichlet, 1805-1859), Riemann em 1854 (Georg Friedrich Bernhard Riemann, 1826-1866), Lipót Fejér em 1904 (Lipót Fejér, 1880-1959) e Lebesgue em 1905 (Henri Léon Lebesgue, 1875-1941), compilados por O'Connor & Robertson (2006). A validade de uma série de Fourier ficou condicionada apenas à integrabilidade da função  $f(\theta)$  (Kanasewich, 1981).

Fourier desenvolveu a idéia para encontrar a solução da equação do calor, como equações diferenciais parciais tendo condições inicial e de contorno (2). A série de Fourier de uma função  $f(\theta)$ , definida no intervalo  $(-l, l)$  é dada por Debnath (2002)<sup>4</sup>:

$$f(\theta) = \sum_{n=-\infty}^{\infty} c_n \exp\left(\frac{in\pi\theta}{l}\right) \tag{3}$$

e os coeficientes de Fourier, que já eram conhecidos para Euler em 1777 (Leonhard Euler, 1707-1783), como as fórmulas integrais:

$$c_n = \frac{1}{2l} \int_{-l}^l f(\theta) \exp\left(-\frac{in\pi\theta}{l}\right) d\theta \tag{4}$$

sendo que  $i = \sqrt{-1}$ . A série de Fourier representa a resolução de uma dada função em um conjunto infinito, mas discreto, de componentes harmônicos. Existe também a

forma contínua da série de Fourier, não apresentada aqui. As séries e integrais de Fourier, e suas aplicações, foram os principais tópicos do tratado publicado em 1822. A questão fundamental da convergência das séries de Fourier foi resolvida por Carleson em 1966 (Lennart Axel Edvard Carleson, 1928-), que provou que essas séries de uma função contínua quase sempre são convergentes (Debnath, 2002).

A transformada de Fourier se originou do teorema da integral de Fourier, estabelecido no seu tratado *Théorie analytique de la chaleur*. As séries e transformadas de Fourier são reconhecidamente uma das mais importantes descobertas da matemática com uma grande gama de aplicações em outros campos científicos. Muitas dessas aplicações fazem uso da transformada de Fourier nos domínios temporais e de frequência. A transformada de Fourier de um sinal, ou função  $f(t)$ , é definida por Debnath (2002):

$$\mathfrak{T}(f(t)) = \hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) \exp(-i\omega t) dt = (f, e^{i\omega t}) \tag{5}$$

Onde  $\hat{f}(\omega)$  é a transformada de Fourier com variável  $\omega$ , ou uma função de frequência  $\omega$  e  $(f, e^{i\omega t})$  é o produto interno num espaço de Hilbert. A transformação de um sinal o decompõe em uma onda de seno de diferentes frequências e fases, sendo freqüentemente chamada de *espectro de Fourier* (Debnath, 2002). Fisicamente, a integral de Fourier mede a oscilação de  $f$  na frequência  $\omega$ , e  $\hat{f}(\omega)$  é chamada de *espectro de frequência* de um sinal ou onda  $f(t)$ . Pode-se referir a  $f(t)$  como a forma da onda no domínio temporal e  $\hat{f}(\omega)$  como a forma de onda no domínio de frequência. Essa terminologia descreve a dualidade e a equivalência das representações das formas de ondas.

O sucesso da transformada de Fourier se deve ao fato que, sob certas condições, o sinal  $f(t)$  pode ser reconstruído pela fórmula da transformada inversa de Fourier:

$$f(t) = \mathfrak{T}^{-1}\{\hat{f}(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) \exp(i\omega t) d\omega = \frac{1}{2\pi} (\hat{f}, e^{-i\omega t}) \tag{6}$$

Segundo Champeney (1987), Fourier não deu precisamente um conjunto de condições sob as quais o teorema da inversão acontecesse. Seu legado se baseia na descoberta e demonstração como o teorema poderia ser útil em aplicações em problemas da física e ficou para matemáticos que lhe sucederam adicionar rigor às suas idéias. Na realidade, o "teorema de Fourier" consiste não apenas de um simples teorema, mas em vários teoremas todos sobre um tema comum.

Para Champeney (1987), muita coisa depende como se interpreta o símbolo da integral nesses teoremas. É necessário decidir entre integração de Lebesgue ou Riemann. Cada interpretação vem acompanhada de seus próprios teoremas. Dessa maneira, há de se reconhecer que a própria definição do que se entende por uma transformada de Fourier torna-se incerto. Os primeiros desenvolvimentos da teoria de Fourier tiveram por base a teoria de integração de Riemann. Uma abordagem alternativa para integração desenvolvida por Lebesgue é mais poderosa e simples de usar para a teoria de Fourier, e virtualmente toda a abordagem da matemática moderna toma por base a teoria de Lebesgue.

<sup>3</sup> Segundo Duddeck (2002), Fourier submeteu seu artigo *On the propagation of heat in solid bodies* ao Instituto de Paris em 21 de dezembro de 1807, tendo sido criticado pelo comitê científico (Lagrange, Laplace, Monge e Lacroix). A objeção, de Lagrange e Laplace, foi sobre a expansão de funções como séries trigonométricas, as conhecidas séries de Fourier. Seu trabalho ficou conhecido a partir de 1822 com a publicação da *Théorie analytique de la chaleur*.

<sup>4</sup> Em termos de seno e cosseno:  $f(t) = \frac{a_0}{2} + \sum_{r=1}^k (a_r \cos rt + b_r \sin rt)$ , sendo:

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) dt, a_r = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos rtdt \text{ e } b_r = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin rtdt$$

Em essência, a diferença entre os processos de integração Lebesgue *versus* Riemann é a seguinte: a integral de Lebesgue trata limites infinitos de integração e certas singularidades infinitas tudo num simples processo de limites, enquanto a integral própria de Riemann não é capaz de tratar com limites infinitos de integração ou com singularidades infinitas na função. Funções com singularidades ou são infinitas ou não têm derivadas finitas em todos os lugares. Singularidade é um ponto onde uma equação, superfície ou função torna-se degenerada (um caso de limites no qual uma classe de objetos muda sua natureza passando a pertencer a outra classe de objetos, normalmente mais simples). Singularidades são também chamadas de pontos singulares.

Como apresentadas em (5) e (6), a transformada de Fourier (TF) e sua inversa (ITF), precisam de duas adaptações para uso no processamento de imagens digitais: trabalhar no plano bidimensional, ou domínio espacial, e com uma versão discreta. A versão da transformada discreta de Fourier (TDF) é dada pela seguinte fórmula de Myler & Weeks (1993):

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[ -i2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (7)$$

onde  $M$  é o número de pixels na direção  $x$ ,  $N$  é o número de pixels na direção  $y$ ,  $u = 0, 1, 2, \dots, M-1$  e  $v = 0, 1, 2, \dots, N-1$ . A transformada inversa de Fourier é dada por Myler & Weeks (1993):

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[ i2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (8)$$

com  $x = 0, 1, 2, \dots, M-1$  e  $y = 0, 1, 2, \dots, N-1$ .

As equações (7) e (8) mostram que a transformada de uma imagem  $M \times N$  produz uma matriz  $M \times N$  de coeficientes. Como a transformada inversa reconstrói a imagem original a partir desse conjunto de coeficientes, eles devem constituir uma representação completa da informação presente na imagem. A transformada de Fourier processa a imagem no *domínio de frequência* e a transformada inversa remete a imagem de volta ao *domínio espacial*. Quando se trabalha com imagens digitais, uma variante das fórmulas acima, chamada transformada rápida de Fourier (TRF), é usada, com base na propriedade da separabilidade (Gonzalez & Wintz, 1977; Lyon, 1999).

## Java Advanced Imaging (JAI)

As classes, ou API, do JAI não fazem parte do pacote padrão do J2SDK, é necessário instalá-las a partir do endereço:

<http://java.sun.com/products/java-media/jai>

além do pacote JAI, o endereço acima contém a documentação sobre a API, endereços de ajuda da comunidade de desenvolvedores do JAI e um tutorial.

O JAI disponibiliza dois operadores fundamentais para se trabalhar no domínio de frequência de Fourier: DFT e IDFT. O operador *discrete Fourier transform* - DFT cria uma imagem no domínio de frequência com dados complexos, onde as bandas pares representam a parte real e as bandas ímpares representam a parte imaginária da transformada de Fourier.

Isso significa dizer que cada pixel no domínio de frequência tem uma magnitude e um ângulo de fase, de maneira que o número de bandas geradas nesse domínio é o dobro do domínio espacial. O mapeamento do domínio de frequência para o domínio espacial é feito usando a função IDFT (*inverse discrete Fourier transform*). Além desses operadores, existem aqueles que trabalham a parte complexa da imagem: Magnitude, MagnitudeSquare, Phase e PeriodicShift.

O operador Magnitude calcula a magnitude de cada pixel de uma imagem complexa, ou espectro de amplitude. A imagem fonte deve ter um número par de bandas (2, 4 etc.), representando a parte real e um número ímpar de bandas (1, 3 etc.) representando a parte imaginária de cada pixel complexo. O operador MagnitudeSquare eleva cada pixel da magnitude ao quadrado, gerando o espectro de potência. A implementação desses operadores é:

```
public PlanarImage imagemMagnitude(PlanarImage dft) {
    System.out.println("Criando imagem magnitude...");
    ParameterBlock pb = new ParameterBlock();
    pb.addSource(dft);
    return JAI.create("magnitude", pb);
}

// Calcula o espectro de potência
public PlanarImage imagemQuadradoMagnitude(PlanarImage dft) {
    System.out.println("Criando quadrado da magnitude...");
    ParameterBlock pb = new ParameterBlock();
    pb.addSource(dft);
    return JAI.create("magnitudeSquared", pb);
}
```

O operador Phase calcula o ângulo de fase de cada pixel de uma imagem complexa. Ele também necessita do conjunto de bandas pares e ímpares, uma imagem DFT. O código de implementação é:

```
public PlanarImage imagemFase(PlanarImage dft) {
    System.out.println("Criando imagem fase...");
    ParameterBlock pb = new ParameterBlock();
    pb.addSource(dft);
    return JAI.create("phase", pb);
}
```

Outro operador de frequência no JAI é o PeriodicShift, que calcula a translação periódica de uma imagem nas direções  $x$  e/ou  $y$ . Esse operador serve para fazer uma translação na origem do espectro de amplitudes, transferindo, por exemplo, a frequência zero para o centro da imagem, um procedimento comum quando se mostra esse espectro:

```
public PlanarImage imagemDeslocaXY(PlanarImage src, int
    shiftX, int shiftY) {
    System.out.println("Calculando translacao periodica...");
    ParameterBlock pb = new ParameterBlock();
    pb.addSource(src);
    pb.add(shiftX);
    pb.add(shiftY);
    return JAI.create("periodicShift", pb);
}
```

Depois que obtem-se as imagens separadas com os espectros de amplitude e fase, pode-se aplicar um filtro de passa-altas ou baixas, ou outra operação, individualmente. Para mostrar o resultado de procedimentos efetuados no domínio de frequência, é necessário agregar novamente as partes real e imaginária, para poder se usar o operador inverso IDFT. O operador `PolarToComplex` realiza esse procedimento, criando uma imagem complexa a partir das imagens de fase e magnitude:

```
// Calcula imagem complexa
public PlanarImage imagemPolar2Complex(PlanarImage mag,
    PlanarImage phase) {
    System.out.println("Criando imagem complexa...");
    ParameterBlock pb = new ParameterBlock();
    pb.addSource(mag);
    pb.addSource(phase);
    return JAI.create("polarTocomplex", pb);
}
```

A `PlanarImage` resultante dessa operação é passada para o IDFT, cuja função é reconstituir a imagem original. A Fig. 1 exemplifica esse processo. O JAI requer um procedimento em cadeia para o operador de Fourier. A figura ilustra o grafo da implementação do programa `JAIIDFT.java`. O objetivo do programa é mostrar a importância da informação contida na componente de fase da imagem original. Se a retirarmos, no caso, tornando-a igual a zero, a recomposição da imagem fica comprometida, como pode ser constatado ao se executar o programa.

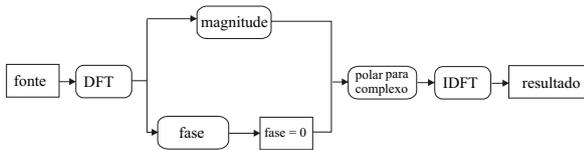


Fig. 1. Grafo do programa `JAIIDFT.java`.

Conforme ilustra o grafo, para se achar a transformada de Fourier de uma imagem, basta passar para DFT o nome da imagem. Esse operador gera duas imagens, representando o espectro de amplitude, ou magnitude, e a fase. Essas duas imagens sintetizam a imagem complexa, gerada por IDFT, no espaço de coordenadas polares,  $(r, \theta)$ , sendo a magnitude a coordenada radial,  $r$ , e a fase, a coordenada angular,  $\theta$ . Para se mapear a imagem do domínio de frequência novamente para o domínio espacial, a imagem deve ser do tipo complexo. O operador `PolarToComplex` realiza esse procedimento, criando a imagem complexa a partir das imagens de fase e magnitude. A partir desse ponto, o operador IDFT é usado para gerar o resultado no domínio espacial.

## Estudo de Caso

A Fig. 2 mostra o resultado da aplicação do programa `JAIIDFT.java` em uma imagem. Quando a informação sobre a fase da imagem é zerada no plano de frequência, a imagem original não pode mais ser recuperada, comprovando que a decomposição da imagem do domínio espacial para o domínio de frequência guardam as mesmas informações, embora em planos diversos.

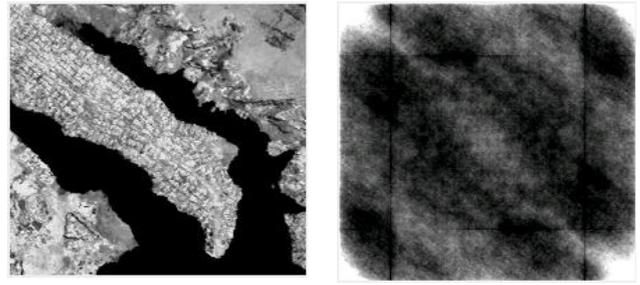


Fig. 2. Resultado do programa `JAIIDFT.java`.

## Conclusões

- A biblioteca de classes do JAI, uma extensão do Java para processamento de imagens digitais, implementa de forma eficiente as funções necessárias para se trabalhar com a transformada de Fourier.
- Várias, e em diversos campos do conhecimento, são as aplicações da transformada de Fourier. As informações aqui fornecidas são suficientes para se começar a usar essa ferramenta, aplicando-a de forma mais indicada ao problema em mão.
- A transformada de Fourier pode ser usada no processo de filtragem de uma imagem digital, substituindo a operação de convolução.
- Dependendo da parte da imagem magnitude em que se aplica um filtro, a imagem pode ser filtrada tanto para deixar passar as baixas ou as altas frequências.
- A implementação Java da transformada de Fourier se encontra operacional e disponível. Para obtenção do fonte, visitar o endereço eletrônico: <http://repositorio.agrolivre.gov.br/projects/pid/>.
- A transformada de Fourier pode ser desenvolvida com a programação Java comum. Nesse caso, fica a cargo do desenvolvedor implementar todos os requisitos necessários para a implementação da transformada. Fica mais fácil usar a extensão oferecida pelo JAI.
- A escolha da biblioteca JAI torna-se, então, a escolha óbvia para quem quer fazer análise de imagens no domínio de frequência usando Java.
- Existem implementações da transformada de Fourier em Java, sem uso do JAI. Qual a melhor implementação? Fica a cargo do usuário fazer um teste comparativo. O efeito líquido da tarefa seria avaliar qual implementação gasta menos tempo para processar imagens digitais de tamanhos diversos.

## Referências Bibliográficas

- CHAMPENEY, D. C. *A handbook of Fourier theorems*. Cambridge, NY: Cambridge University Press, 1987. 198 p.
- DEBNATH, L. *Wavelet transforms and their applications*. Boston: Birkhäuser, 2002. 588 p.
- DUDDECK, F. M. E. *Fourier BEM: generalization of boundary element methods by Fourier transform*. Berlin: Springer-Verlag, 2002. 183 p.

FOURIER, J. B. J. *The analytical theory of heat*. Mineola, NY: Courier Dover, 2003. 496 p.

GONZALEZ, R. C.; WINTZ, P. *Digital image processing*. Reading, MA: Addison-Wesley, 1977. 431 p.

HOWELL, K. B. *Principles of Fourier analysis*. Boca Raton: CRC Press, 2001. 792 p.

JAYAKUMAR, R. *Method of applied Fourier analysis*. Boston: Birkhäuser, 1998. 385 p.

KANASEWICH, E. R. *Time sequence analysis in geophysics*. Alberta: University of Alberta, 1981. 480 p.

LYON, D. *Image processing in Java*. Upper Saddle River: Prentice Hall, 1999. 532 p.

MYLER, H. R.; WEEKS, A. R. *Computer imaging in C*. Englewood Cliffs: Prentice Hall, 1993. 284 p.

O'CONNOR, J. J.; ROBERTSON, E. F. *MacTutor history of Mathematics*. Disponível em: <<http://www-history.mcs.st-andrews.ac.uk/history/index.html>>. Acesso em: 05 out. 2006.

PRESTINI, E. *The evolution of applied harmonic analysis: models of the real world*. Boston: Birkhäuser, 2003. 368 p.

## Comunicado Técnico, 74

Ministério da Agricultura, Pecuária e Abastecimento



Embrapa Informática Agropecuária  
Área de Comunicação e Negócios (ACN)  
Endereço: Caixa Postal 6041 - Barão Geraldo  
13083-970 - Campinas, SP  
Fone: (19) 3789-5743  
Fax: (19) 3289-9594  
e-mail: sac@cnptia.embrapa.com.br

1ª edição on-line - 2006

© Todos os direitos reservados.

## Comitê de Publicações

**Presidente:** Kleber Xavier Sampaio de Souza.  
**Membros Efetivos:** Adriana Farah Gonzalez (secretária), Ivanilde Dispatto, José Iguelmar Miranda, Marcia Izabel Fugisawa Souza, Silvio Roberto Medeiros Evangelista, Stanley Robson de Medeiros Oliveira.

**Suplentes:** Laurimar Gonçalves Vandrúsculo, Maria Goretti Gurgel Praxedes.

## Expediente

**Supervisor editorial:** Ivanilde Dispatto  
**Normalização bibliográfica:** Marcia Izabel Fugisawa Souza  
**Editoração eletrônica:** Área de Comunicação e Negócios