em que *f* 0 é a **resolução de frequência** [separação entre amostras de *G*(*f*)], caso *f* 0 seja dado, podemos selecionar o valor de *T* 0

segundo a Eq. (3.106). Conhecidos os valores de *T* 0 e *T s* , determinamos o de *N* 0 de



Em geral, se o sinal for limitado no tempo, *G*(*f*) não será limitado em frequência e haverá mascaramento no cálculo de *Gq* . Para reduzir o efeito de mascaramento, precisamos aumentar a frequência de dobramento, ou seja, devemos reduzir o valor de *T s* (intervalo de amostragem) tanto quanto praticamente possível. Se o sinal for limitado em frequência, *g*(*t*) não será limitado no tempo, de modo que haverá mascaramento (sobreposição) no cálculo de *g k* . Para reduzir este mascaramento, precisamos aumentar o valor de *T* 0 , o período de *g k* . Isso implica a redução do intervalo *f* 0 (em hertz) de amostragem em frequência. Em qualquer dos casos (redução de *T s* , no caso de sinal limitado no tempo, ou aumento de *T* 0 , no caso de sinal limitado em frequência), para maior precisão, precisamos aumentar o número de amostras *N* 0 , pois *N* 0 = *T* 0 /*T s* . Existem, ainda, sinais que não são limitados nem no tempo nem em frequência. Para estes sinais, devemos reduzir *T s* e aumentar *T* 0 .

## Pontos de Descontinuidade

Caso *g*(*t*) tenha, em um ponto de amostragem, uma descontinuidade do tipo degrau, o valor da amostra deve ser tomado como a média dos valores nos dois lados da descontinuidade, pois a representação de Fourier em um ponto de descontinuidade converge para o valor médio.

## Uso de Algoritmo de FFT no Cálculo de DFT

O número de contas necessário para o cálculo de uma DFT foi drasticamente reduzido em um algoritmo desenvolvido por Tukey e Cooley, em 1965. 6 Esse algoritmo, conhecido como **transformada de Fourier rápida**\* (**FFT** — *fast Fourier transform*), reduz o número de contas de algo da ordem de *N* 2 0 para *N* 0 log *N* 0 . Para calcular o valor de uma amostra *Gr* pela Eq. (3.103a), precisamos de *N* 0 multiplicações complexas e *N* 0 − 1 adições complexas. Para calcular *N* 0 valores de *Gr* (*r* = 0, 1, ..., *N* 0 − 1), precisamos de um total de *N* 2 0 multiplicações complexas e *N* 0 (*N* 0 − 1) adições complexas. Para grandes valores de *N* 0 , isto pode exigir um tempo proibitivamente grande, mesmo com o uso de computador de alta velocidade. O algoritmo de FFT é um salva­vidas em aplicações de processamento de sinais. O algoritmo de FFT fica simplificado se escolhermos *N* 0 como uma potência de 2, embora isto não seja, em geral, necessário. Detalhes da FFT podem ser encontrados em qualquer livro sobre processamento de sinais (por exemplo, Ref. 3).

3.10 EXERCÍCIOS COM O MATLAB

## Cálculo de Transformadas de Fourier

Nesta seção de exercícios baseados em computador, consideremos dois exemplos para ilustrar o uso de DFT no cálculo da transformada de Fourier. Usaremos MATLAB para calcular a DFT com o algoritmo de FFT. No primeiro exemplo, o sinal é *g*(*t*) = *e*

−2*t u*(*t*), com início em *t* = 0, e no segundo, *g*(*t*) = Π(*t*), com início em *t* = −½.

EXEMPLO COMPUTACIONAL C3.1

Empreguemos a DFT (implementada pelo algoritmo de FFT) para calcular a transformada de Fourier de *e* −2*t u*(*t*) e, a seguir, tracemos o gráfico do resultante espectro de Fourier.

Primeiro, devemos determinar *T s* e *T* 0 . A transformada de Fourier de *e* −2*t u*(*t*) é 1/(2*πf* + 2). Esse sinal passa­faixa não é limitado em frequência. Tomemos sua largura de banda essencial como a frequência em que |*G*(*f*)| se torna igual a 1% do valor de pico, que ocorre em *f* = 0. Observemos que



O pico de |*G*(*f*)| ocorre em *f* = 0, em que |*G*(0)| = 0,5. Portanto, a largura de banda essencial *B* corresponde a *f = B*, com



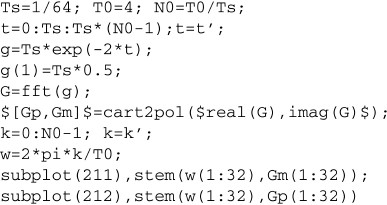
e, da Eq. (3.105b),



Arredondemos esse valor para *T s* = 0,015625 segundo, de modo que tenhamos 64 amostras por segundo. Agora, devemos determinar *T* 0 . O sinal não é limitado no tempo. Precisamos truncá­lo em *T* 0 , tal que *g*(*T* 0 ) ≪ 1. Escolhamos *T* 0 = 4 (oito constantes de tempo do sinal), o que resulta em *N* 0 = *T* 0 /*T s* = 256, que é uma potência de 2. Vale ressaltar que há

muita flexibilidade na determinação de *T s* e *T* 0 , dependendo da precisão desejada e da capacidade computacional disponível. Poderíamos ter escolhido *T* 0 = 8 e *T s* = 1/32, o que também resultaria em *N* 0 = 256, mas implicaria um erro de mascaramento ligeiramente maior.

Como o sinal tem uma descontinuidade do tipo degrau em *t* = 0, o valor da primeira amostra (em *t* = 0) é 0,5, média dos valores nos dois lados da descontinuidade. O programa de MATLAB que implementa a DFT com o algoritmo de FFT é o seguinte:



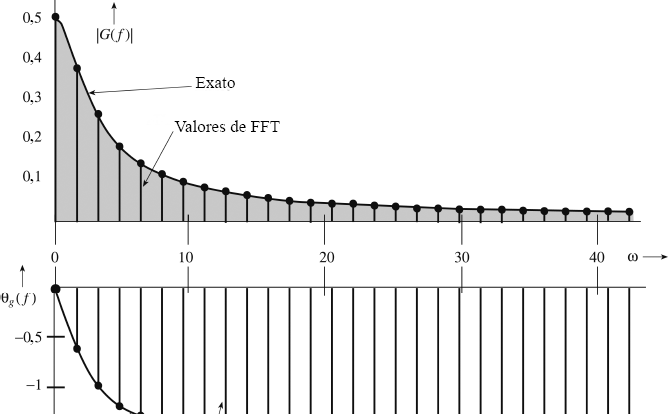
Como *Gq* tem período *N* 0 , *Gq = G* (*q*+256) , de modo que *G* 256 = *G* 0 . Portanto, basta traçar o gráfico de *Gq* no intervalo *q*

= 0 a *q* = 255 (e não 256). Além disso, devido à periodicidade, *G* −*q* = *G* (−*q*+256) , ou seja, os valores de *Gq* no intervalo *q* =

−127 a *q* = −1 são idênticos aos valores de *Gq* no intervalo *q* = 129 a *q* = 255. Logo, *G* −127 = *G* 129 , *G* −126 = *G* 130 ,..., *G* −1 = *G* 255 . Adicionalmente, devido à propriedade de simetria conjugada da transformada de Fourier, *G* −*q* = *G\* q* ; assim, *G* 129 = *G*\* 127 , *G* 130 = *G*\* 126 ,..., *G* 255 = *G*\* 1 . Consequentemente, para sinais de valores reais, não é necessário marcar no gráfico os valores de *Gq* com *q* maior que *N* 0 /2 (128, neste caso), pois são os complexos conjugados dos valores de *Gq* com *q* = 0 a 128.

O gráfico do espectro de Fourier na Fig. 3.40 mostra amplitude e fase das amostras de *G*(*f*) tomadas em intervalos de 1/*T* 0

= 1/4 Hz, ou *ω* 0 = 1,5708 rad/s. Na Fig. 3.40, mostramos apenas os primeiros 28 pontos (em vez dos 128 pontos), para evitar o acúmulo excessivo de dados no gráfico.





**Figura 3.40** Transformada de Fourier discreta de um sinal exponencial *e* −2*t u*(*t*). O eixo horizontal é *ω* (em radianos por segundo).

Neste exemplo, dispúnhamos da expressão analítica de *G*(*f*), o que nos permitiu fazer escolhas INTELIGENTES para *B* (ou frequência de amostragem *f s* ). Na prática, em geral, não conhecemos *G*(*f*). Na verdade, isso é exatamente o que desejamos calcular. Nesses casos, para determinar *B* ou *f s* , devemos lançar mão de evidências circunstanciais. Devemos, sucessivamente, reduzir o valor de *T s* e calcular a transformada até que o resultado satisfaça o desejado número de algarismos significativos.

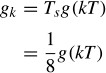
A seguir, calcularemos a transformada de Fourier de *g*(*t*) = 8 Π(*t*).

EXEMPLO COMPUTACIONAL C3.2

Empreguemos a DFT (implementada pelo algoritmo de FFT) para calcular a transformada de Fourier de 8 Π(*t*) e tracemos o gráfico do resultante espectro de Fourier.

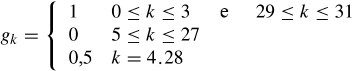
Essa função retangular e sua transformada de Fourier são mostradas na Fig. 3.41a e b. Para determinar o valor do intervalo de amostragem *T s* , devemos, primeiro, definir a largura de banda essencial *B*. Da Fig. 3.41b, vemos que *G*(*f*) decai lentamente com *f*. Consequentemente, a largura de banda essencial *B* é bastante grande. Por exemplo, em *B* = 15,5 Hz (97,39 rad/s), *G*(*f*) = −0,1643, o que corresponde a cerca de 2% do valor de pico, *G*(0). Poderíamos, então, tomar a largura de banda essencial como 16 Hz. No entanto, deliberadamente, tomaremos *B* = 4 Hz, por dois motivos: (1) mostrar o efeito de mascaramento e (2) o uso de *B* > 4 implicaria enorme número de amostras, que não poderiam ser mostradas de forma adequada em uma página de livro sem perda de detalhes fundamentais. Portanto, aceitaremos a aproximação para que possamos esclarecer conceitos de DFT por meio de gráficos.

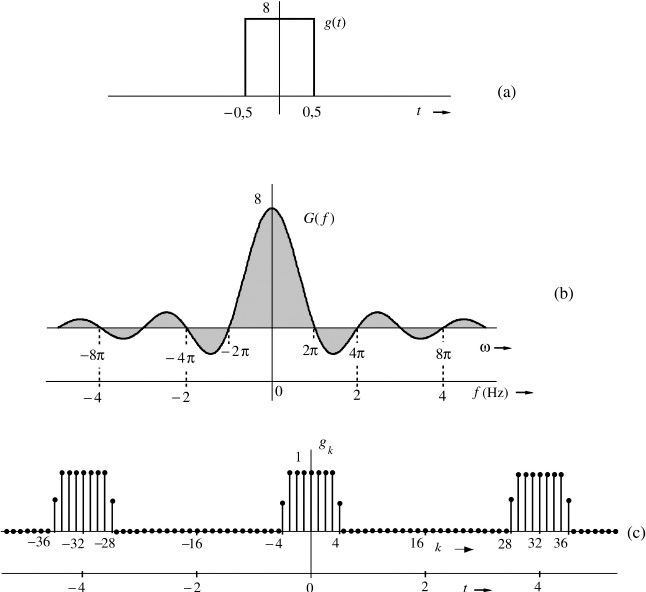
A escolha *B* = 4 resulta em um intervalo de amostragem *T s* = 1/2*B* = 1/8 segundos. Examinando novamente o espectro na Fig. 3.41b, vemos que a escolha da resolução de frequência *f* 0 = 1/4 Hz é razoável, e corresponde a quatro amostras em cada lóbulo de *G*(*f*). Neste caso, *T* 0 = 1/*f* 0 = 4 segundos, e *N* 0 = *T* 0 /*T s* = 32. A duração de *g*(*t*) é de apenas 1 segundo. Devemos repetir *g*(*t*) a cada 4 segundos, como indicado na Fig. 3.41c, e tomar amostras a cada 0,125 segundo. Isso nos dará 32 amostras (*N* 0 = 32). Também temos

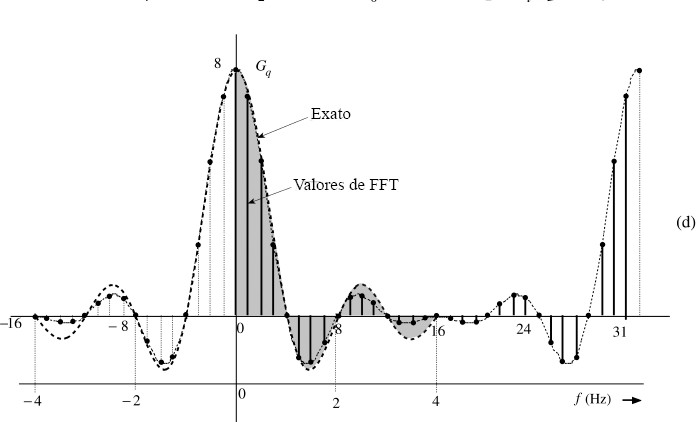


Como *g*(*t*) = 8 Π(*t*), os valores de *g k* são 1, 0 ou 0,5 (nos pontos de descontinuidade), como mostrado na Fig. 3.41c; nessa figura, por conveniência, *g k* é mostrado como função de *t* e de *k*.

Na dedução da DFT, supomos que *g*(*t*) tem início em *t* = 0 (Fig. 3.39a) e tomamos *N* 0 amostras no intervalo (0, *T* 0 ). No caso em consideração, contudo, *g*(*t*) tem início em *t* = −½. Essa dificuldade é facilmente resolvida quando observamos que a DFT obtida por este procedimento é, na verdade, a DFT de *g k* repetido a cada *T* 0 segundos. Da Fig. 3.41c, fica claro que a repetição periódica do segmento de *g k* no intervalo de −2 a 2 segundos é equivalente à repetição do segmento de *g k* no intervalo de 0 a 4 segundos. Portanto, a DFT das amostras colhidas entre −2 e 2 segundos é igual à DFT das amostras colhidas entre 0 e 4 segundos. Assim, independentemente do instante em que *g*(*t*) tem início, sempre podemos tomar as amostras de *g*(*t*) e repetilas periodicamente no intervalo de 0 a *T* 0 . No presente exemplo, os valores das 32 amostras são





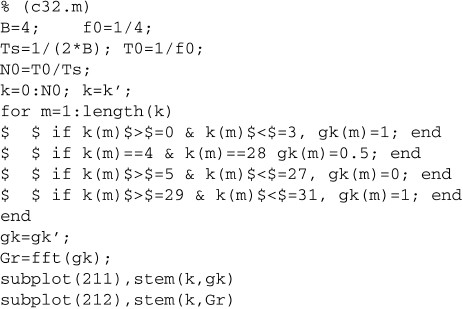


**Figura 3.41** Transformada de Fourier discreta de um pulso retangular.

Vale ressaltar que a última amostra é tomada em *t* = 31/8 e não em *t* = 4, pois a repetição do sinal reinicia em *t* = 4, de modo que a amostra em *t* = 4 é igual à amostra em *t* = 0. Com *N* 0 = 32, Ω 0 = 2*π*/32 = *π*/16. Logo, [ver a Eq. (3.103a)],



O programa MATLAB que usa o algoritmo de FFT para calcular a DFT é dado a seguir. Primeiro, escrevemos um programa MATLAB para gerar 32 amostras de *g k* e, então, calculamos a DFT.



A Fig. 3.41d mostra o gráfico de *Gq* .

As amostras *Gq* são espaçadas de *f* 0 = 1/*T* 0 Hz. Neste exemplo, *T* 0 = 4 segundos, de modo que a resolução de frequência *f* 0 é ¼ Hz, como desejado. A frequência de dobramento *f s* /2 = *B* = 4 Hz corresponde a *q = N* 0 /2 = 16. Como *Ga* tem período *N* 0 (*N* 0 = 32), os valores de *Gq* para *q* entre −16 e −1 são iguais àqueles para *q* entre 16 e 31. A DFT nos fornece amostras do espectro *G*(*f*).

Para facilitar a comparação, a Fig. 3.41d também mostra a curva hachurada 8 sinc(π*f*), que é a transformada de Fourier de 8 Π(*t*). Os valores de *Gq* calculados pela DFT exibem erro de mascaramento, o que fica claro quando comparamos os

dois gráficos. O erro em *G* 2 é da ordem de apenas 1,3%. No entanto, o erro de mascaramento aumenta rapidamente com *r*. Por exemplo, o erro em *G* 6 é de cerca de 12%, e o erro em *G* 10 , 33%. O erro em *G* 14 é de assustadores 72%. O erro percentual aumenta de forma muito rápida nas proximidades da frequência de dobramento (*r* = 16), pois *g*(*t*) tem uma descontinuidade degrau, o que faz com que *G*(*f*) decaia muito lentamente, como 1/*f*. Assim, nas proximidades da frequência de dobramento, a cauda invertida (devido ao mascaramento) é quase igual a *G*(*f*). Além disso, os valores extremos são a diferença entre os valores exato e da parte que sofreu dobra (quase iguais aos exatos). Consequentemente, o erro percentual nas proximidades da frequência de dobramento (*r* = 16, neste exemplo) é muito alto, embora o erro absoluto seja muito pequeno. Fica claro que, para sinais com descontinuidades do tipo degrau, o erro de mascaramento nas proximidades da frequência de dobramento sempre será grande (em termos percentuais), qualquer que seja o valor escolhido para *N* 0 . Para garantir erro de mascaramento desprezível para qualquer valor de *q*, devemos assegurar que *N* 0

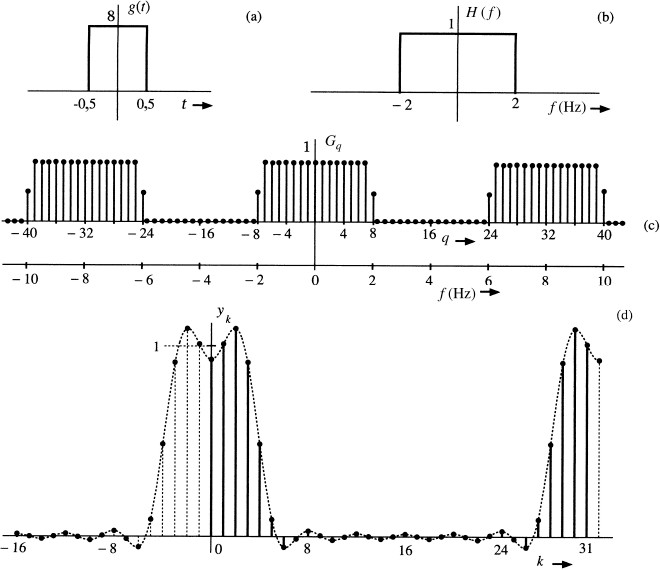
≫ *q*. Essa observação se aplica a todos os sinais com descontinuidade do tipo degrau.

## Filtragem

Quando pensamos em filtragem, em geral, o fazemos em termos de uma solução orientada a hardware (ou seja, montagem de um circuito com componentes *RLC* e amplificadores operacionais). Contudo, a filtragem também admite uma solução orientada a software [algoritmo computacional que fornece a saída filtrada *y*(*t*), para uma dada entrada *g*(*t*)]. Isso pode ser implementado de modo conveniente via DFT. Seja *g*(*t*) o sinal a ser filtrado; então, os valores *Gq* , DFT de *g k* , são calculados. O espectro *Gq* é formatado (filtrado) como desejado através da multiplicação de *Gq* por *Hq* , em que *Hq* são as amostras da função de transferência do filtro, *H*(*f*) [*Hq = H*(*qf* 0 )]. Por fim, calculamos a DFT inversa (ou IDFT) de *Gq Hq* e obtemos a saída filtrada *y k* [*y k* = *T s y*(*kT*)]. O próximo exemplo ilustra este procedimento.

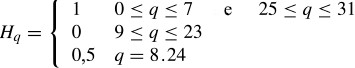
EXEMPLO COMPUTACIONAL C3.3

O sinal *g*(*t*) na Fig. 3.42a é aplicado a um filtro passa­baixos ideal, cuja função de transferência *H*(*f*) é mostrada na Fig. 3.42b. Usemos a DFT para calcular a saída do filtro.



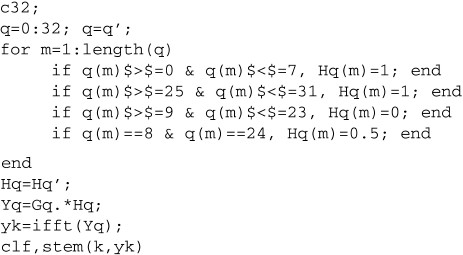
**Figura 3.42** Filtragem de *g*(*t*) por *H*(*f*).

Já calculamos a DFT de *g*(*t*) com 32 amostras (Fig. 3.41d). Agora, devemos multiplicar *Gq* por *Hq* . Para calcular *Hq* , recordemos que, na determinação da DFT de *g*(*t*) com 32 amostras, usamos *f* 0 = 0,25 Hz. Como *Gq* tem período *N* 0 = 32, *Hq* deve ter o mesmo período e, portanto, amostras espaçadas de 0,25 Hz. Isso significa que *Hq* deve se repetir a cada 8 Hz ou 16*π* rad/s (ver Fig. 3.42c). Assim, as 32 amostras de *Hq* são produzidas, no intervalo 0 ≤ *f* ≤ 8, como

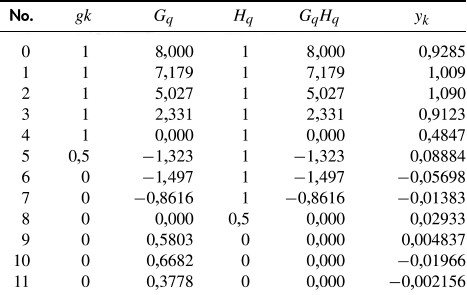


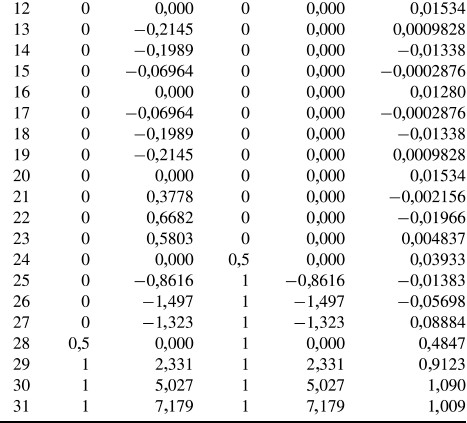
Multiplicamos *Gq* por *Hq* e calculamos a DFT inversa. O resultante sinal de saída é mostrado na Fig. 3.42d. A Tabela 3.4 lista valores de *g k* , *Gq* , *Hq* , *Y q* e *y k* .

No Exemplo C.32, já calculamos a DFT de *g*(*t*) com 32 amostras (*Gq* ). O programa MATLAB do Exemplo C3.2 pode ser armazenado como um arquivo.m (por exemplo, “c32.m”). Podemos importar *Gq* no ambiente MATLAB via comando “c32”. A seguir, geramos 32 amostras de *Hq* , multiplicamos *Gq* por *Hq* e, para obter *y k* , calculamos a DFT inversa. Também podemos obter *y k* calculando a convolução de *g k* e *h k* .



**Tabela 3.4**





# REFERÊNCIAS

1. R. V. Churchill and J. W. Brown, *Fourier Series and Boundary Value Problems*, 3rd ed., McGraw­Hill, New York, 1978.
2. R. N. Bracewell, *Fourier Transform and Its Applications*, rev. 2nd ed., McGraw­Hill, New York, 1986.
3. B. P. Lathi, *Signal Processing and Linear Systems*, Oxford University Press, 2000.
4. E. A. Guillemin, *Theory of Linear Physical Systems*, Wiley, New York, 1963.
5. F. J. Harris, “On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform,” *Proc. IEEE*, vol. 66, pp. 51– 83, Jan. 1978.
6. J. W. Tukey and J. Cooley, “An Algorithm for the Machine Calculation of Complex Fourier Series,” *Mathematics of Computation*, Vol. 19, pp. 297–301, April 1965.

# EXERCÍCIOS

**3.1­1**

Mostre que a transformada de Fourier de *g*(*t*) pode ser expressa como



A seguir, mostre que, caso *g*(*t*) seja uma função par de *t*,



e, caso *g*(*t*) seja uma função ímpar de *t*,



Agora, mostre que:

*Se g*(*t*) *for*: *Então G*(*f*) *é:*