Integral Transforms

Deepak U. Patil and Madhu N Belur
deepakp@ee.iitb.ac.in
belur@ee.iitb.ac.in

Dept. of Electrical Engineering
Indian Institute of Technology, Bombay
Funded by National Mission on Education through ICT

December 23, 2009

Calculus/Laplace Transform Using Symbolic Toolbox

Fourier Transforms

System Administration Tasks

- Interface between Maxima and Scilab.

System Administration Tasks

- Interface between Maxima and Scilab.
- Enter directory where toolbox is kept. You will see two directories.

System Administration Tasks

- Interface between Maxima and Scilab.
- Enter directory where toolbox is kept. You will see two directories.
- Execute builder.sce and loader.sce (from both OVLD and SYM directories).

System Administration Tasks

- Interface between Maxima and Scilab.
- Enter directory where toolbox is kept. You will see two directories.
- Execute builder.sce and loader.sce (from both OVLD and SYM directories).
- In our case Just type `exec('symbolic.sce')`.

System Administration Tasks

- Interface between Maxima and Scilab.
- Enter directory where toolbox is kept. You will see two directories.
- Execute builder.sce and loader.sce (from both OVLD and SYM directories).
- In our case Just type `exec('symbolic.sce')`.
- `restartserver` and `killserver`

System Administration Tasks

- Interface between Maxima and Scilab.
- Enter directory where toolbox is kept. You will see two directories.
- Execute builder.sce and loader.sce (from both OVLD and SYM directories).
- In our case Just type `exec('symbolic.sce')`.
- `restartserver` and `killserver`
- Create Symbolic Objects 'a' and 'b' using `syms a b`.

System Administration Tasks

- Interface between Maxima and Scilab.
- Enter directory where toolbox is kept. You will see two directories.
- Execute builder.sce and loader.sce (from both OVLD and SYM directories).
- In our case Just type `exec('symbolic.sce')`.
- `restartserver` and `killserver`
- Create Symbolic Objects 'a' and 'b' using `syms a b`.
- Symbolic Operations can be done with these objects.

Symbolic Differentiation/Integration/Limits

- `diff(f,x)` //differentiate function f w.r.t x.

Symbolic Differentiation/Integration/Limits

- `diff(f,x)` //differentiate function f w.r.t x.
- `integ(f,x)` // Indefinite Integral of f w.r.t x.

Symbolic Differentiation/Integration/Limits

- `diff(f,x)` //differentiate function f w.r.t x.
- `integ(f,x)` // Indefinite Integral of f w.r.t x.
- `integ(f,x,0,a)`
  //Definite Integral of f w.r.t x from limits 0 to a(a is numeric constant).

Symbolic Differentiation/Integration/Limits

- `diff(f,x)` //differentiate function f w.r.t x.
- `integ(f,x)` // Indefinite Integral of f w.r.t x.
- `integ(f,x,0,a)`
  //Definite Integral of f w.r.t x from limits 0 to a(a is numeric constant).
- `limit(f,x,0)` //limit of f as x tends to 0.

Laplace Transforms

- Laplace Transform: $F(s) = \int_0^\infty f(t)e^{-st}dt$

Laplace Transforms

- Laplace Transform: $F(s) = \int\limits_0^\infty f(t)e^{-st}dt$
- Create Symbolic Objects 's' and 't'.

Laplace Transforms

- Laplace Transform: $F(s) = \int_0^\infty f(t)e^{-st}dt$
- Create Symbolic Objects 's' and 't'.
- `laplace(p)` //laplace transform of function p.

Laplace Transforms

- Laplace Transform: $F(s) = \int\limits_{0}^{\infty} f(t)e^{-st}dt$
- Create Symbolic Objects 's' and 't'.
- `laplace(p)` //laplace transform of function p.
- `ilaplace(p)`//Inverse Laplace Transform of function p.

Laplace Transforms

- Laplace Transform: $F(s) = \int\limits_0^\infty f(t)e^{-st}dt$
- Create Symbolic Objects 's' and 't'.
- `laplace(p)` //laplace transform of function p.
- `ilaplace(p)`//Inverse Laplace Transform of function p.
- e.g. `syms s t`
  `laplace(t)`
  `ilaplace(1/s^2)`

Preliminary Version

- There are a lot of bugs.

Preliminary Version

- There are a lot of bugs.
- Command are similar to Matlab Symbolic toolbox.

Preliminary Version

- There are a lot of bugs.
- Command are similar to Matlab Symbolic toolbox.
- All function of its matlab counterpart are not available.

Preliminary Version

- There are a lot of bugs.
- Command are similar to Matlab Symbolic toolbox.
- All function of its matlab counterpart are not available.
- One can always depend on help documentation.

Preliminary Version

- There are a lot of bugs.
- Command are similar to Matlab Symbolic toolbox.
- All function of its matlab counterpart are not available.
- One can always depend on help documentation.
- Reference:
  www.cert.fr/dcsd/idco/perso/Magni/s_sym/doc/index.html

Fourier Transform

- The continuous Fourier Transform is defined as: $F(\omega) = \int\limits_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$
- We will consider Discrete Fourier Transform (DFT).
- DFT is defined as $X(p) = \sum_{n=0}^{N-1} f(n)e^{\frac{-2\pi pnj}{N}}$

Discrete Fourier Transform

For a periodic sequence: DFT (Discrete Fourier Transform) gives the frequency content.
Linear transformation on the input sequence.
Take signal values of just one period: finite dimensional signal (due to periodicity of N).

$$X(k) := \sum_{n=0}^{N-1} x(n) e^{\frac{-2\pi ik}{N} n} \text{ for } k = 0, \ldots, N-1 \text{ (analysis equation)}$$

$e^{-2\pi ikN}$ is the N^{th} root of unity.
Inverse DFT for the synthesis equation. Normalization constants vary in the literature.

Discrete Fourier Transform

What is the matrix defining relating the DFT $X(k)$ of the signal $x(n)$? Define $\omega := e^{\frac{-2\pi ik}{N}n}$.

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2N-2} \\ 1 & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2N-2} & \cdots & \omega^{(N-1)^2} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

(Note: $\omega^N = 1$, etc.)
Check that the above $N \times N$ matrix has nonzero determinant.
(Change of basis.) Moreover, columns are orthogonal.
Orthonormal? (Normalization (by $\sqrt{N}$) not done yet.)

Discrete Fourier Transform and interpolation

Van der monde matrix: closely related to interpolation problems

Discrete Fourier Transform and interpolation

Van der monde matrix: closely related to interpolation problems
Of course, inverse DFT is nothing but interpolation! Used in computation of determinant of a polynomial matrix.
Construct $p(s) := x_0 + x_1 s + x_2 s^2 \cdots + x_{N-1} s^{N-1}$

Discrete Fourier Transform and interpolation

Van der monde matrix: closely related to interpolation problems
Of course, inverse DFT is nothing but interpolation! Used in computation of determinant of a polynomial matrix.
Construct $p(s) := x_0 + x_1 s + x_2 s^2 \cdots + x_{N-1} s^{N-1}$
To obtain $X(k)$, evaluate p at $s = \omega^k$.
$X(k) = p(\omega^k)$

Discrete Fourier Transform and interpolation

Van der monde matrix: closely related to interpolation problems
Of course, inverse DFT is nothing but interpolation! Used in computation of determinant of a polynomial matrix.
Construct $p(s) := x_0 + x_1 s + x_2 s^2 \cdots + x_{N-1} s^{N-1}$
To obtain $X(k)$, evaluate p at $s = \omega^k$.
$X(k) = p(\omega^k)$ `horner command`
`Given values of` $p(\omega^k)$ `for various` ω^k `(i.e.,` $X(k)$`),` `find the coefficients of the polynomial` $p(s)$`: inverse DFT: interpolation of a polynomial to ‘fit’ given values at specified (complex) numbers.`

FFT

Since many powers of ω are repeated in that matrix (only $N-1$ powers are different, many real/imaginary parts are repeated for even N), redundancy can be drastically decreased.

FFT

Since many powers of ω are repeated in that matrix (only $N-1$ powers are different, many real/imaginary parts are repeated for even N), redundancy can be drastically decreased.
Length of the signal is a power of 2: recursive algorithm possible.

FFT: recursive implementation

- Separate $p(s)$ (coefficients $x_0, \ldots, x_{N-1}$) into its even and odd powers (even and odd indices k). N is divisible by 2.
- Compute DFT of p_{odd} and p_{even} separately. (Do same separation, if possible.)
- Let X_{odd} and X_{even} denote the individual DFT's. (Same length.)
- Define $D := \mathrm{diag}(1, \omega, \omega^2, \ldots, \omega^{\frac{N}{2}-1})$
- Combine the two separate DFT's using the formula

$$X(k) = X_{\mathrm{even}} + DX_{\mathrm{odd}} \text{ for } k = 0, \ldots, \frac{N}{2} - 1$$
$$X(k) = X_{\mathrm{even}} - DX_{\mathrm{odd}} \text{ for } k = \frac{N}{2}, \ldots, N - 1$$

FFT using Scilab Command

- `fft`
  //Calculates the DFT of given signal using Fast Fourier Transform Algorithm
- `ifft` //Inverse DFT can be obtained.
- e.g. Discretize $f = \sin t$ by putting `t=0:N-1` into N samples.
  `F=fft(f) //DFT of vecor f`
  `f=ifft(F) //inverse DFT of F.`