DOWNLOAD PDF FOURIER ANALYSIS OF TIME SERIES

Chapter 1 : Fourier series - Wikipedia

Fourier Analysis of Time Series. by Dr. R. L. Herman, UNC Wilmington. Friday, September 20, This is a work in progress. Introduction. Often one is interested in determining the frequency content of signals. Signals are typically represented as time dependent functions. Real signals are continuous, or analog signals.

And just to remind you of our basic strategy, essentially, the idea was to exploit the notion of linearity by decomposing the input into a sum of basic inputs and then using linearity to tell us that the output can be represented as the corresponding linear combination of the associated outputs. So, if we have a linear system, either continuous-time or discrete-time, for example, with continuous time, if the input is decomposed as a linear combination of basic inputs, with each of these basic inputs generating an associated output, and if the system is linear, then the output of the system is the same linear combination of the associated outputs. And the same statement is identical both for continuous time and discrete time. So the strategy is to decompose the input into these basic inputs. And the inputs were chosen also with some particular strategy in mind. In particular, for both continuous time or discrete time, in this representation, the basic inputs used in the decomposition are chosen, first of all, so that a broad class of signals could be represented in terms of these basic inputs, and second of all, so that the response to these basic inputs is, in some sense, easy to compute. Now, in the representation which led us to convolution, the particular choice that we made in the discrete-time case for our basic inputs was a decomposition of the input in terms of delayed impulses. And the associated outputs that that generated were delayed versions of the impulse response. Decomposing the input into a linear combination of these, the output into the corresponding linear combination of these, then led to the convolution sum in the discrete time case. And in the continuous-time case, a similar kind of decomposition, in terms of impulses, and associated representation of the output, in terms of the impulse response, led to the convolution integral. Now, the basic strategy, of course, requires that we choose a set of inputs, basic building blocks, which have two properties. One is that the system response be straightforward to compute, or in some sense, easy to compute. And second is that it be a fairly general set of building blocks so that we can build lots of signals out of them. So for now, we want to restrict ourselves to complex exponentials of a particular form, and in fact, also initially to continuous-time signals and systems. Now more or less, the justification for this, or the proof, follows by simply looking at the response to a complex exponential, using the convolution integral. We can then recognize that this complex exponentials can be factored into two terms. And so we can rewrite this complex exponential as this product. Second, recognize that this term can be taken outside the integral, over here, because of the fact that it depends only on t and not on Tau. And so, in fact, we put in a complex exponential, we get out a complex exponentials of the same frequency, multiplied by a complex constant. So in fact, this function is the Eigenfunction. And this value is the Eigenvalue. Namely you put it through the system, they come out with the same form and simply scale. And in fact, that turns out to be the case with complex exponentials. So the first thing we want to deal with are periodic signals and the Fourier series. And the Fourier series is a representation for periodic continuous-time signals. We have a signal, then, which is periodic. So this is the fundamental frequency. So as k, an integer, varies, these correspond to harmonically related complex exponentials. And this summation is what will be referred to as the Fourier series. And as we proceed with the discussion, there are two issues that will develop. And the second question will be how broad a class of signals, in fact, can be represented this way. In other words, this index k runs over limits that include both negative values and positive values. Now that complex exponential form is one representation for the Fourier series. Or another equivalent trigonometric form involves rearranging this in terms of a combination of cosines and sines. Now in this representation, the frequencies of the sinusoids vary only over positive frequencies. And typically one thinks of periodic signals as having positive frequencies associated with them. So the representation that we will most typically be using is the complex exponential form. The algebraic steps are ones that you can pursue more leisurely just to kind of verify them and step through them. The integral of a of a periodic of a sinusoid, cosine or sine, over an integral number of periods is 0. And the integral of the sine is equal to 0. And then essentially what happens at this point, algebraically, is that we use the result that we

just developed to evaluate this integral. So multiplying both sides of the Fourier series and then doing the integration leads us, after the appropriate manipulation, to the expression that I have up here. And the upshot of all that, then, is that the right hand side of this expression disappears except for the term.

Chapter 2 : python - Using fourier analysis for time series prediction - Stack Overflow

A new, revised edition of a yet unrivaled work on frequencydomain analysis. Long recognized for his unique focus on frequency domain methodsfor the analysis of time series data as well as for his applied,easy-to-understand approach, Peter Bloomfield brings his well-known work thoroughly up to date.

The human ear automatically and involuntarily performs a calculation that takes the intellect years of mathematical education to accomplish. The ear formulates a transform by converting soundâ€”the waves of pressure traveling over time and through the atmosphereâ€”into a spectrum, a description of the sound as a series of volumes at distinct pitches. The brain then turns this information into perceived sound. A similar conversion can be done using mathematical methods on the same sound waves or virtually any other fluctuating signal that varies with respect to time. The Fourier transform is the mathematical tool used to make this conversion. Simply stated, the Fourier transform converts waveform data in the time domain into the frequency domain. The Fourier transform accomplishes this by breaking down the original time-based waveform into a series of sinusoidal terms, each with a unique magnitude, frequency, and phase. This process, in effect, converts a waveform in the time domain that is difficult to describe mathematically into a more manageable series of sinusoidal functions that when added together, exactly reproduce the original waveform. Plotting the amplitude of each sinusoidal term versus its frequency creates a power spectrum, which is the response of the original waveform in the frequency domain. The Fourier transform has become a powerful analytical tool in diverse fields of science. In some cases, the Fourier transform can provide a means of solving unwieldy equations that describe dynamic responses to electricity, heat or light. In other cases, it can identify the regular contributions to a fluctuating signal, thereby helping to make sense of observations in astronomy, medicine and chemistry. Perhaps because of its usefulness, the Fourier transform has been adapted for use on the personal computer. Algorithms have been developed to link the personal computer and its ability to evaluate large quantities of numbers with the Fourier transform to provide a personal computer-based solution to the representation of waveform data in the frequency domain. But what should you look for in Fourier analysis software? What makes one software package better than another in terms of features, flexibility, and accuracy? This application note will present and explain some of the elements of such software packages in an attempt to remove the mystery surrounding this powerful analytical tool. All graphics and concepts presented in this note are also derived from the WWB Fourier transform utility. A Trio of Transforms Before computers, numerical calculation of a Fourier transform was a tremendously labor intensive task because such a large amount of arithmetic had to be performed with paper and pencil. These calculations became more practical as computers and programs were developed to implement new methods of Fourier analysis. One such method was developed in by James W. Cooley and John W. The fast Fourier transform FFT is a computationally efficient method of generating a Fourier transform. The main advantage of an FFT is speed, which it gets by decreasing the number of calculations needed to analyze a waveform. A disadvantage associated with the FFT is the restricted range of waveform data that can be transformed and the need to apply a window weighting function to be defined to the waveform to compensate for spectral leakage also to be defined. The DFT allows you to precisely define the range over which the transform will be calculated, which eliminates the need to window. The transformation from the time domain to the frequency domain is reversible. Once the power spectrum is displayed by one of the two previously mentioned transforms, the original signal can be reconstructed as a function of time by computing the inverse Fourier transform IFT. Each of these transforms will be discussed individually in the following paragraphs to fill in missing background and to provide a yardstick for comparison among the various Fourier analysis software packages on the market. For example, calculated directly, a DFT on 1, i. But the increase in speed comes at the cost of versatility. The FFT function automatically places some restrictions on the time series to be evaluated in order to generate a meaningful, accurate frequency response. Because the FFT function uses a base 2 logarithm by definition, it requires that the range or length of the time series to be evaluated contains a total number of data points precisely equal to a 2-to-the-nth-power number e. Therefore, with an FFT you can only

evaluate a fixed length waveform containing points, or points, or points, etc. For example, if your time series contains data points, you would only be able to evaluate of them at a time using an FFT since is the highest 2-to-the-nth-power that is less than Because of this 2-to-the-nth-power limitation, an additional problem materializes. When a waveform is evaluated by an FFT, a section of the waveform becomes bounded to enclose points, or points, etc. One of these boundaries also establishes a starting or reference point on the waveform that repeats after a definite interval, thus defining one complete cycle or period of the waveform. Any number of waveform periods and more importantly, partial waveform periods can exist between these boundaries. This is where the problem develops. Obviously, the chances of a waveform containing a number of points equal to a 2-to-the-nth-power number and ending on a whole number of periods are slim at best, so something must be done to ensure an accurate representation in the frequency domain. This waveform possesses end-point continuity as shown in c , which means the resulting power spectrum will be accurate and no window need be applied. A more typical encounter is shown in b , where the range of the FFT does not contain a whole number of periods. The discontinuity in the end-points of this waveform d means the resulting power spectrum will contain high frequency components not present in the input, requiring the application of a window to attenuate the discontinuity and improve accuracy. Think of the length of waveform to be evaluated as a ring that has been uncoiled. Thus, the FFT would evaluate this waveform with the end-point error and generate a power spectrum containing false frequency components representative of the end-point mismatch. This figure shows the power spectrum of two sine waves of equal amplitude and frequency. However, the peak of the right power spectrum appears somewhat "spread out". This inaccuracy is the result of an FFT performed on a waveform that does not contain a whole number of periods. The spreading out or "leakage" effect of the right power spectrum is due to energy being artificially generated by the discontinuity at the end points of the waveform. Fortunately, a solution exists to minimize this leakage effect error and ensure accuracy in the frequency domain. Aside from the DFT to be defined , the only solution is to multiply the time series by a window weighting function before the FFT is performed. However, if your waveform has important information appearing at the ends of the window, it will be destroyed by the tapering. In this case, a solution other than a window must be sought. With the window approach, the periodically incorrect signal as processed by the FFT will have a smooth transition at the end points which results in a more accurate power spectrum representation. A number of windows exist. Each has different characteristics that make one window better than the others at separating spectral components near each other in frequency, or at isolating one spectral component that is much smaller than another, or whatever the task. Some popular windows named after their inventors are Hamming, Bartlett, Hanning, and Blackman. The Hamming window offers the familiar bell-shaped weighting function but does not bring the signal to zero at the edges of the window. The Hamming window produces a very good spectral peak, but features only fair spectral leakage reduction. The Bartlett window offers a triangular shaped weighting function that brings the signal to zero at the edges of the window. This window produces a good, sharp spectral peak and is good at reducing spectral leakage as well. The Hanning window offers a similar bell-shaped window a good approximation to the shape of the Hanning window can be seen inFigure 5d that also brings the signal to zero at the edges of the window. The Hanning window produces good spectral peak sharpness as good as the Bartlett window , but the Hanning offers very good spectral leakage reduction better than the Bartlett. The Blackman window offers a weighting function similar to the Hanning but narrower in shape. Because of the narrow shape, the Blackman window is the best at reducing spectral leakage, but the trade-off is only fair spectral peak sharpness. It depends upon your skill at manipulating the trade-offs between the various window constraints and also on what you want to get out of the power spectrum or its inverse. Obviously, a Fourier analysis software package that offers a choice of several windows is desirable to eliminate spectral leakage distortion inherent with the FFT. In short, the FFT is a computationally fast way to generate a power spectrum based on a 2-to-the-nth-power data point section of waveform. This means that the number of points plotted in the power spectrum is not necessarily as many as was originally intended. The FFT also uses a window to minimize power spectrum distortion due to end-point discontinuity. With these limitations inherent to the FFT, does the Fourier analysis software package you are considering offer a solution other than the FFT? Another solution abandons windowing in favor of

allowing the user to precisely define the range over which the Fourier transform will be calculated. This approach nullifies the 2-to-the-nth-power limitation and is called a DFT. For example, if you are processing transient signals, the edges contain important information that will be unacceptably distorted by applying the window solution. In this case, you would have no choice but to use the DFT. As stated previously, the DFT allows you to adjust the end-points that define the range of the waveform to be transformed, thus eliminating the need for windowing. This approach allows a waveform containing any number of points to be evaluated, which provides more flexibility than the fixed-length, 2-to-the-nth-power FFT. However, to prevent the same leakage effect experienced with a non-windowed FFT, the DFT must be generated over a whole number of periods starting at the waveforms mean level crossing. In other words, the end-points that define the range of the waveform over which the DFT will be calculated must be adjusted to enclose or define a whole number of periods, preferably starting at or around the point where the waveform crosses its mean. However, versatility and precision come at the expense of added computation time by the algorithm and added time spent by you on end-point positioning. The times shown are in seconds and were obtained from a based, 25 megahertz PC without a math coprocessor. Since the WWB Fourier transform algorithm uses integer arithmetic, a math co-processor does little to increase performance and is therefore not needed for this package. Some software packages either require a math co-processor for operation or strongly recommend one for optimal performance. If the FFT is performed on a fractional number of periods, the spectrum gives a very different picture as shown on the bottom in the illustration above â€” a broad peak resulting in poorly determined frequency and inaccurate amplitude. These waveforms were generated by an inexpensive function generator, which accounts for the noise present in the spectrum. More typically, b shows the transform of the same waveform only with mismatched end-points. Note that the second peak is not even visible in this spectrum. The need for a window clearly exists. The remaining transforms illustrate the degree of success attained with various windows in suppressing spectral leakage and recovering the lost frequency component. Each window was applied to the original waveform, with the result illustrating the trade-off between sharpness of peaks and decay of sidelobes. Note that this window never brings the signal to zero. For this spectral-separation example, the Blackman window is the best at bringing out the weaker term as a well defined peak.

Chapter 3 : Fourier Series introduction (video) | Khan Academy

Fourier analysis grew from the study of Fourier series, and is named after Joseph Fourier, who showed that representing a function as a sum of trigonometric functions greatly simplifies the study of heat transfer. Today, the subject of Fourier analysis encompasses a vast spectrum of mathematics.

Fourier analysis has many scientific applications â€" in physics , partial differential equations , number theory , combinatorics , signal processing , digital image processing , probability theory , statistics , forensics , option pricing , cryptography , numerical analysis , acoustics , oceanography , sonar , optics , diffraction , geometry , protein structure analysis, and other areas. This wide applicability stems from many useful properties of the transforms: The transforms are usually invertible. The exponential functions are eigenfunctions of differentiation , which means that this representation transforms linear differential equations with constant coefficients into ordinary algebraic ones Evans Therefore, the behavior of a linear time-invariant system can be analyzed at each frequency independently. By the convolution theorem , Fourier transforms turn the complicated convolution operation into simple multiplication, which means that they provide an efficient way to compute convolution-based operations such as polynomial multiplication and multiplying large numbers Knuth The discrete version of the Fourier transform see below can be evaluated quickly on computers using Fast Fourier Transform FFT algorithms. The FT method is used to decode the measured signals and record the wavelength data. And by using a computer, these Fourier calculations are rapidly carried out, so that in a matter of seconds, a computer-operated FT-IR instrument can produce an infrared absorption pattern comparable to that of a prism instrument. For example, JPEG compression uses a variant of the Fourier transformation discrete cosine transform of small square pieces of a digital image. The Fourier components of each square are rounded to lower arithmetic precision , and weak components are eliminated entirely, so that the remaining components can be stored very compactly. In image reconstruction, each image square is reassembled from the preserved approximate Fourier-transformed components, which are then inverse-transformed to produce an approximation of the original image. Applications in signal processing[edit] When processing signals, such as audio , radio waves , light waves, seismic waves , and even images, Fourier analysis can isolate narrowband components of a compound waveform, concentrating them for easier detection or removal. A large family of signal processing techniques consist of Fourier-transforming a signal, manipulating the Fourier-transformed data in a simple way, and reversing the transformation. Equalization of audio recordings with a series of bandpass filters ; Digital radio reception without a superheterodyne circuit, as in a modern cell phone or radio scanner ; Image processing to remove periodic or anisotropic artifacts such as jaggies from interlaced video, strip artifacts from strip aerial photography , or wave patterns from radio frequency interference in a digital camera; Cross correlation of similar images for co-alignment; X-ray crystallography to reconstruct a crystal structure from its diffraction pattern; Fourier transform ion cyclotron resonance mass spectrometry to determine the mass of ions from the frequency of cyclotron motion in a magnetic field; Many other forms of spectroscopy, including infrared and nuclear magnetic resonance spectroscopies; Generation of sound spectrograms used to analyze sounds; Passive sonar used to classify targets based on machinery noise. The relative computational ease of the DFT sequence and the insight it gives into S f make it a popular analysis tool. Continuous Fourier transform[edit] Main article: Fourier transform Most often, the unqualified term Fourier transform refers to the transform of functions of a continuous real argument, and it produces a continuous function of frequency, known as a frequency distribution. One function is transformed into another, and the operation is reversible. When the domain of the input initial function is time t , and the domain of the output final function is ordinary frequency , the transform of function s t at frequency f is given by the complex number:

Chapter 4 : fft - Perform Fourier Analysis to a Time Series in R - Stack Overflow

Fourier Analysis of Time Series: An Introduction. Peter Bloomï¬•eld Copyright John Wiley & Sons, Inc. ISBN: Created Date: 5/1/ PM.

Media In mathematics , Fourier analysis is the study of the way general functions may be represented or approximated by sums of simpler trigonometric functions. Fourier analysis grew from the study of Fourier series , and is named after Joseph Fourier , who showed that representing a function as a sum of trigonometric functions greatly simplifies the study of heat transfer. Today, the subject of Fourier analysis encompasses a vast spectrum of mathematics. In the sciences and engineering, the process of decomposing a function into oscillatory components is often called Fourier analysis, while the operation of rebuilding the function from these pieces is known as Fourier synthesis. For example, determining what component frequencies are present in a musical note would involve computing the Fourier transform of a sampled musical note. One could then re-synthesize the same sound by including the frequency components as revealed in the Fourier analysis. In mathematics, the term Fourier analysis often refers to the study of both operations. The decomposition process itself is called a Fourier transformation. Its output, the Fourier transform, is often given a more specific name, which depends on the domain and other properties of the function being transformed. Moreover, the original concept of Fourier analysis has been extended over time to apply to more and more abstract and general situations, and the general field is often known as harmonic analysis. Each transform used for analysis see list of Fourier-related transforms has a corresponding inverse function transform that can be used for synthesis. Fourier analysis has many scientific applications â€" in physics , partial differential equations, number theory , combinatorics , signal processing, digital image processing, probability theory, statistics , forensics , option pricing , cryptography , numerical analysis, acoustics , oceanography , sonar , optics , diffraction , geometry , protein structure analysis, and other areas. This wide applicability stems from many useful properties of the transforms: The transforms are usually invertible. The exponential functions are of derivative , which means that this representation transforms linear differential equations with constant coefficients into ordinary algebraic ones. Therefore, the behavior of a LTI system can be analyzed at each frequency independently. By the convolution theorem, Fourier transforms turn the complicated convolution operation into simple multiplication, which means that they provide an efficient way to compute convolution-based operations such as polynomial multiplication and multiplying large numbers. The discrete version of the Fourier transform see below can be evaluated quickly on computers using Fast Fourier Transform FFT algorithms. In forensics, laboratory infrared spectrophotometers use Fourier transform analysis for measuring the wavelengths of light at which a material will absorb in the infrared spectrum. The FT method is used to decode the measured signals and record the wavelength data. And by using a computer, these Fourier calculations are rapidly carried out, so that in a matter of seconds, a computer-operated FT-IR instrument can produce an infrared absorption pattern comparable to that of a prism instrument. Fourier transformation is also useful as a compact representation of a signal. For example, JPEG compression uses a variant of the Fourier transformation discrete cosine transform of small square pieces of a digital image. The Fourier components of each square are rounded to lower arithmetic precision, and weak components are eliminated entirely, so that the remaining components can be stored very compactly. In image reconstruction, each image square is reassembled from the preserved approximate Fourier-transformed components, which are then inverse-transformed to produce an approximation of the original image. Applications in signal processing When processing signals, such as Sound , , light waves, , and even images, Fourier analysis can isolate narrowband components of a compound waveform, concentrating them for easier detection or removal. A large family of signal processing techniques consist of Fourier-transforming a signal, manipulating the Fourier-transformed data in a simple way, and reversing the transformation. Equalization of audio recordings with a series of ; Digital radio reception without a superheterodyne circuit, as in a modern cell phone or radio scanner ; Image processing to remove periodic or anisotropic artifacts such as jaggies from interlaced video, strip artifacts from strip aerial photography, or wave patterns from radio frequency interference in a digital camera; Cross correlation of similar images for

co-alignment; X-ray crystallography to reconstruct a crystal structure from its diffraction pattern; Fourier transform ion cyclotron resonance mass spectrometry to determine the mass of ions from the frequency of cyclotron motion in a magnetic field; Many other forms of spectroscopy, including infrared and nuclear magnetic resonance spectroscopies; Generation of sound used to analyze sounds; Passive sonar used to classify targets based on machinery noise. Continuous Fourier transform Most often, the unqualified term Fourier transform refers to the transform of functions of a continuous real number argument, and it produces a continuous function of frequency, known as a frequency distribution. One function is transformed into another, and the operation is reversible. When the domain of the input initial function is time , and the domain of the output final function is frequency , the transform of function at frequency is given by the complex number: Evaluating this quantity for all values of produces the frequency-domain function. Then can be represented as a recombination of complex exponentials of all possible frequencies: The complex number, , conveys both amplitude and phase of frequency. See Fourier transform for much more information, including: The Fourier transform of a periodic function, , with period , becomes a Dirac comb function, modulated by a sequence of complex coefficients: The inverse transform, known as Fourier series, is a representation of in terms of a summation of a potentially infinite number of harmonically related sinusoids or complex exponential functions, each with an amplitude and phase specified by one of the coefficients: When , is expressed as a periodic summation of another function,: See Fourier series for more information, including the historical development. Thus, a convergent periodic summation in the frequency domain can be represented by a Fourier series, whose coefficients are samples of a related continuous time function: We may also note that: The Fourier series coefficients and inverse transform , are defined by: Parameter corresponds to the sampling interval, and this Fourier series can now be recognized as a form of the Poisson summation formula. Thus we have the important result that when a discrete data sequence, , is proportional to samples of an underlying continuous function, , one can observe a periodic summation of the continuous Fourier transform,. That is a cornerstone in the foundation of digital signal processing. Furthermore, under certain idealized conditions one can theoretically recover and exactly. A sufficient condition for perfect recovery is that the non-zero portion of be confined to a known frequency interval of width. When that interval is , the applicable reconstruction formula is the Whittakerâ€”Shannon interpolation formula. Another reason to be interested in is that it often provides insight into the amount of aliasing caused by the sampling process. Applications of the DTFT are not limited to sampled functions. See Discrete-time Fourier transform for more information on this and other topics, including:

Chapter 5 : FFT (Fast Fourier Transform) Waveform Analysis

The Fourier Transform (FFT) â€¢Based on Fourier Series - represent periodic time series data as a sum of sinusoidal components (sine and cosine).

This gives us the discrete transform pair Note that this is similar to the definition of the FFT given in Matlab. There are various implementations of it, but a standard form is the Radix-2 FFT. We describe this FFT in the current section. We begin by writing the DFT compactly using. Note that and We can then write The key to the FFT is that this sum can be written as two similar sums: For even For odd we have Each of these equations gives the Fourier coefficients in terms of a similar sum using fewer terms and with a different weight, IF N is a power of 2, then this process can be repeated over and over until one ends up with a simple sum. The process is easily seen when written out for a small number of samples. Let Then a first pass at the above gives The point is that the terms in these expressions can be regrouped with and noting: This takes 8 additions and 4 multiplications. There are three stages, amounting to a total of 12 multiplications and 24 additions. Carrying out the process in general, one has steps with multiplications and N additions per step. In the direct computation one has multiplications and additions. Thus, for that would be 49 multiplications and 56 additions. The above process is typically shown schematically in a "butterfly diagram". Examples can be found at other sites. One might be provided here at a later date. Output and Binary Representation Desired Order The binary representation of the index was also listed. Notice that the output is in bit-reversed order as compared to the right side of the table which shows the coefficients in the correct order. The basic trigonometric identities that one needs the product of trigonometric functions expanded as simple expressions. These are based upon the sum and difference identities: Adding or subtracting one obtains. Another frequently occurring computation is the sum of a geometric progression. This is a sum of the form. This is a sum of terms in which consecutive terms have a constant ratio, The sum is easily computed. One multiplies the sum by and subtracts the resulting sum from the original sum to obtain Factoring on both sides of this chain of equations yields the desired sum, The beauty of using Matlab is that many operations can be performed using matrix operations and that one can perform complex arithmetic. This eliminates many loops and make the coding of computations quicker. However, one needs to be able to understand the formalism. In this section we elaborate on these operations so that one can see how the Matlab implementation of the direct computation of the DFT can be carried out in compact form as shown previously in the Matlab Implementation section. A key operation between matrices is matrix multiplication. An matrix is simply a collection of numbers arranged in n rows and m columns. For example, the matrix is a matrix. The entries elements of a general matrix A can be represented as which represents the ith row and jth column. Given two matrices, A and B, we can define the multiplication of these matrices when the number of columns of A equals the number of rows of B. The product, which we represent as matrix C, is given by the ijth element of C. In particular, we let A be a matrix and B an matrix. We can let a and b be and matrices, respectively. Then the product would be a matrix; namely, the sum we are seeking. However, these matrices are not always of the suggested size. A matrix is called a row vector and a matrix is called a column vector. Often we have that both are of the same type. One can convert a row vector into a column vector, or vice versa, using the matrix operation called a transpose. More generally, the transpose of a matrix is defined as follows: Thus, if we want to perform the above sum, we have. In particular, if both a and b are row vectors, the sum in Matlab is given by and if they are both row vectors, the sum is This notation is much easier to type. In our computation of the DFT, we have many sums. For example, we want to compute the coefficients of the sine functions, The sum can be computed as a matrix product. The function y only has values at times This is the sampled data. We can represent it as a vector. The sine functions take values at arguments angles depending upon p and n. So, we can represent the sines as an or matrix. The Fourier coefficient thus becomes a simple matrix multiplication, ignoring the prefactor. Thus, if we put the sampled data in a vector Y and put the sines in an vector S, the Fourier coefficient will be the product, which has size. The A coefficients are computed in the same manner. Comparing the two codes in that section, we see how much easier it is to implement. However, the number of multiplications and additions

has not decreased. This is why the FFT is generally better. But, seeing the direct implementation helps one to understand what is being computed before seeking a more efficient implementation, such as the FFT.

Chapter 6 : Fourier analysis of time series: an introduction - Peter Bloomfield - Google Books

I would like to perform fourier transform to a time series using R. I would like to: Get the sum of the 5th to 18th harmonics; plot each wave; and output as a csv file.

Chapter 7 : Fourier Analysis

For data that is known to have seasonal, or daily patterns I'd like to use fourier analysis be used to make predictions. After running fft on time series data, I obtain coefficients.

Chapter 8 : Fourier analysis - Wikipedia

The discrete-time Fourier transform is a periodic function, often defined in terms of a Fourier series. The Z-transform, another example of application, reduces to a Fourier series for the important case |z|=1.

Chapter 9 : Fourier Analysis of Time Series

Fourier Analysis of Time Series. If there are physical reasons to think that a time series of data is stationary, then Fourier analysis of the data can lead to a number of powerful techniques useful in applications. One begins the analysis by taking the finite-length segment of data in the sequence and estimating the Fourier coefficients for representing the data as a Fourier series on the segment.