**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 28 May 2020 | **Name:** | Anupama J S |
| **Course:** | DSP | **USN:** | 4AL16EC005 |
| **Topic:** | 1. Intuition of Fourier Transform and Laplace Transform 2. Laplace Transform of First order 3. Implementation of Laplace Transform using Matlab 4. Applications of Z-Transform 5. Find the Z-Transform of sequence using Matlab | **Semester & Section:** | 8th sem “A”section |
| **Github Repository:** | AnupamaJS |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **INTUITION OF FOURIER TRANSFORM AND LAPLACE TRANSFORM**  **C:\Users\User\Downloads\WhatsApp Image 2020-05-28 at 9.08.32 PM.jpeg**  An Introduction to Laplace Transforms and Fourier Series will be useful for second and third year undergraduate students in engineering, physics or mathematics, as well as for graduates in any discipline such as financial mathematics, econometrics and biological modelling requiring techniques for solving initial value  One main point of the Fourier transform is that it exchanges differentiation and multiplication  by t, under suitable hypotheses  f^’(x) = ix ˆf(x).  We saw an application of this when we computed the Fourier transform of e−t2/2. In principle one can use this to solve differential equations, as we saw in Example 9. However this does not work so well in many cases, mainly because the functions that solve our equations do not fulfil our ’ suitable hypotheses’. If for example we try to solve a very simple equation  f’= f  or even  f’=0  we get only f = 0 as solutions. This is because the solutions to these equations, f = cet and f = c, are not in L1 so they have no Fourier transforms. The Laplace transform is a way around this difficulty. It operates on functions defined only on the interval [0,∞), that do not grow faster than exponentially.  Definition 2. Let f(t) be a function defined on [0,∞) which satisfies an estimate|f(t)| ≤ AeBt for some constants A and B. Then its Laplace transform is the function  ˜f(s) = L(f)(s) =  It is defined for complex numbers s such that Re s > B; then the integral in the definition is convergent. First of all we note that the Laplace transform determines the function uniquely - if we know the Laplace tranform we can in principle compute the function.  **APPLICATIONS OF Z-TRANSFORM**  C:\Users\User\Downloads\WhatsApp Image 2020-05-28 at 9.08.32 PM (1).jpeg  In mathematics and signal processing, the Z-transform converts a discrete-time signal, which is a sequence of real or complex numbers, into a complex frequency-domain representation.  It can be considered as a discrete-time equivalent of the Laplace transform. This similarity is explored in the theory of time-scale calculus.  The basic idea now known as the Z-transform was known to [Laplace](https://en.wikipedia.org/wiki/Laplace), and it was re-introduced in 1947 by [W. Hurewicz](https://en.wikipedia.org/wiki/Witold_Hurewicz) and others as a way to treat sampled-data control systems used with radar. It gives a tractable way to solve linear, constant-coefficient [difference equations](https://en.wikipedia.org/wiki/Difference_equation). It was later dubbed "the z-transform" by [Ragazzini](https://en.wikipedia.org/wiki/John_R._Ragazzini" \o "John R. Ragazzini) and [Zadeh](https://en.wikipedia.org/wiki/Lotfi_A._Zadeh" \o "Lotfi A. Zadeh) in the sampled-data control group at Columbia University in 1952.  The modified or [advanced Z-transform](https://en.wikipedia.org/wiki/Advanced_Z-transform) was later developed and popularized by [E. I. Jury](https://en.wikipedia.org/wiki/Eliahu_I._Jury).  The idea contained within the Z-transform is also known in mathematical literature as the method of [generating functions](https://en.wikipedia.org/wiki/Generating_function) which can be traced back as early as 1730 when it was introduced by [de Moivre](https://en.wikipedia.org/wiki/Abraham_de_Moivre) in conjunction with probability theory. From a mathematical view the Z-transform can also be viewed as a [Laurent series](https://en.wikipedia.org/wiki/Laurent_series) where one views the sequence of numbers under consideration as the (Laurent) expansion of an analytic function.  The Z-transform can be defined as either a one-sided or two-sided transform.[8]  **Bilateral Z-transform**  The bilateral or two-sided Z-transform of a discrete-time signal {\displaystyle x[n]}x[n] is the formal power series {\displaystyle X(z)}X(z) defined as  **Unilateral Z-transform**  Alternatively, in cases where {\displaystyle x[n]}x[n] is defined only for {\displaystyle n\geq 0}n\geq 0, the single-sided or unilateral Z-transform is defined as In signal processing, this definition can be used to evaluate the Z-transform of the unit impulse response of a discrete-time causal system. An important example of the unilateral Z-transform is the probability-generating function, where the component {\displaystyle x[n]}x[n] is the probability that a discrete random variable takes the value {\displaystyle n}n, and the function {\displaystyle X(z)}X(z) is usually written as {\displaystyle X(s)}X(s) in terms of {\displaystyle s=z^{-1}}{\displaystyle s=z^{-1}}. The properties of Z-transforms (below) have useful interpretations in the context of probability theory. |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 28 May 2020 | **Name:** | Anupama J S |
| **Course:** | Python | **USN:** | 4AL16EC005 |
| **Topic:** | 1. Interacting with databases | **Semester & Section:** | 8th sem “A”section |
| **Github Repository:** | AnupamaJS |  |  |
| **AFTERNOON SESSION DETAILS** | | | |
| **DATABASE PROGRAMMING IN PYTHON**  **C:\Users\User\Downloads\WhatsApp Image 2020-05-28 at 8.15.26 PM.jpeg** | | | |
| From a construction firm to a stock exchange, every organisation depends on large databases. These are essentially collections of tables, and’ connected with each other through columns. These database systems support SQL, the Structured Query Language, which is used to create, access and manipulate the data. SQL is used to access data, and also to create and exploit the relationships between the stored data. Additionally, these databases support database normalisation rules for avoiding redundancy of data. The Python programming language has powerful features for database programming. Python supports various databases like MySQL, Oracle, Sybase, PostgreSQL, etc. Python also supports Data Definition Language (DDL), Data Manipulation Language (DML) and Data Query Statements. For database programming, the Python DB API is a widely used module that provides a database application programming interface.  Benefits of Python for database programming  There are many good reasons to use Python for programming database applications:   * Programming in Python is arguably more efficient and faster compared to other languages. * Python is famous for its portability. * It is platform independent. * Python supports SQL cursors. * In many programming languages, the application developer needs to take care of the open and closed connections of the database, to avoid further exceptions and errors. In Python, these connections are taken care of. * Python supports relational database systems. * Python database APIs are compatible with various databases, so it is very easy to migrate and port database application interfaces.   **SQLITE PYTHON: INSERTING DATA**  **C:\Users\User\Downloads\WhatsApp Image 2020-05-28 at 8.15.26 PM (2).jpeg**  Summary: in this tutorial, you will learn how to insert rows into a table in the SQLite database from a Python program using the sqlite3 module.  To insert rows into a table in SQLite database, you use the following steps:  First, connect to the SQLite database by creating a Connection object.  Second, create a Cursor object by calling the cursor method of the Connection object.  Third, execute an INSERT statement. If you want to pass arguments to the INSERT statement, you use the question mark (?) as the placeholder for each argument.  SQLite Python – inserting rows example  Let’s insert a new project into the projects table and some tasks into the tasks table that we created in the creating tables from a Python program tutorial.  Python SQLite Sample Database  First, create a new function to establish a database connection to an SQLitte database specified by the database file.  def create\_connection(db\_file):  """ create a database connection to the SQLite database  specified by db\_file  :param db\_file: database file  :return: Connection object or None  """  conn = None  try:  conn = sqlite3.connect(db\_file)  except Error as e:  print(e)  return conn  Next, develop a function to insert a new project into the projects table.  def create\_project(conn, project):  """  Create a new project into the projects table  :param conn:  :param project:  :return: project id  """  sql = ''' INSERT INTO projects(name,begin\_date,end\_date)  VALUES(?,?,?) '''  cur = conn.cursor()  cur.execute(sql, project)  return cur.lastrowid  In this function, we used the lastrowid attribute of the Cursor object to get back the generated id.  **PYTHON POSTGRESQL USING PSYCOPG2**  **C:\Users\User\Downloads\WhatsApp Image 2020-05-28 at 8.15.26 PM (1).jpeg**  This Python PostgreSQL tutorial demonstrates how to develop Python database applications with the PostgreSQL database server. In Python, we have serval modules available to connect and work with PostgreSQL. the following are the list.  Psycopg2  pg8000  py-postgresql  PyGreSQL  ocpgdb  bpgsql  SQLAlchemy. SQLAlchemy needs any of the above to be installed separately.  Note: Above all interfaces or modules are adhere to Python Database API Specification v2.0 (PEP 249). This API has been designed to encourage and maintain the similarity between the Python modules that are used to access databases. In other words, the syntax, method and the way of access database are the same in all the modules  Psycopg2 is the most popular python driver for PostgreSQL.  It is required for most Python and Postgres frameworks.  Actively maintained and support the major version of python i.e. Python 3 and Python 2.  It is thread-safe (threads can share the connections). It was designed for heavily multi-threaded applications.  This is mainly focuses on: –  Installing Psycopg2 and use its API to access the PostgreSQL database.  It then takes you through data insertion, data retrieval, data update, and data deletion.  Next, it will cover transaction management, connection pooling, and error-handling techniques to develop robust python programs with PostgreSQL.  **Install Psycopg2 using the pip command**  You need to install Psycopg2 on your machine to use PostgreSQL from Python. This module is available on pypi.org.  Using pip command, you can install Psycopg2 on any operating system including Windows, macOS, Linux, and Unix and Ubuntu. Use the following pip command to install Psycopg2.  pip install psycopg2  You can also install a specific version using the following command.  pip install psycopg2=2.7.5  If you are facing pip install error like “connection error: [SSL: CERTIFICATE\_VERIFY\_FAILED] certificate verify failed (\_ssl.c:598)”. You can resolve this error by setting pypi.org and files.pythonhosted.org as trusted hosts. Please try following the pip command to install Psycopg2.  python -m pip install --trusted-host | | | |