XMUT220 Signals and Systems

Laboratory 6: z-Transforms

Due Date: Not for marking (no submission required)

A. z-Transform

In short, z-transform is the discrete-time counterpart to the Laplace transform. By using this more general representation of discrete-time signals, we are able to obtain characterisation of discrete-time LTI systems and their interactions with signals that is possible beyond the discrete-time Fourier transform (DTFT). For example, DTFT could be applied only to stable LTI systems as DTFT exists only if the impulse response is convergence.

In z-transform, the output of an LTI system is obtained by multiplying the z-transform of the input by the z-transform of the impulse response. We define the z-transform of the impulse response as the transfer function of a system. The transfer function generalises the frequency response characterisation of a system's input-output behaviour and offer new insights into system characteristics.

We will derive the z-transform by examining the effect of applying a complex exponential input to an LTI system. Let $z = re^{j\Omega}$ be a complex number with magnitude r and angle Ω. The signal $x[n] = z^n$ is a complex exponential signal. We use $z = re^{j\Omega}$ to write:

$$x[n] = r^n \cos(\Omega n) + jr^n \sin(\Omega n) \qquad (Eq. 1)$$

As illustrated in the figure below, the real part of $x[n]$ is an exponentially damped cosine and the imaginary part is an exponentially damped sine. The positive number r determine the damping factor and Ω is the sinusoidal frequency. Note that $x[n]$ is a complex sinusoid if $r = 1$.

Consider applying $x[n]$ to an LTI system with impulse response $h[n]$. The system output is given by:

$$y[n] = H\{x[n]\} = h[n] * x[n] = \sum_{k=-\infty}^{\infty} h[k]\, x[n-k] \qquad (Eq. 2)$$

We use $x[n] = z^n$ to obtain:

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]\, z^{n-k} = z^n \left(\sum_{k=-\infty}^{\infty} h[k]\, z^{-k} \right) \qquad (Eq. 3)$$

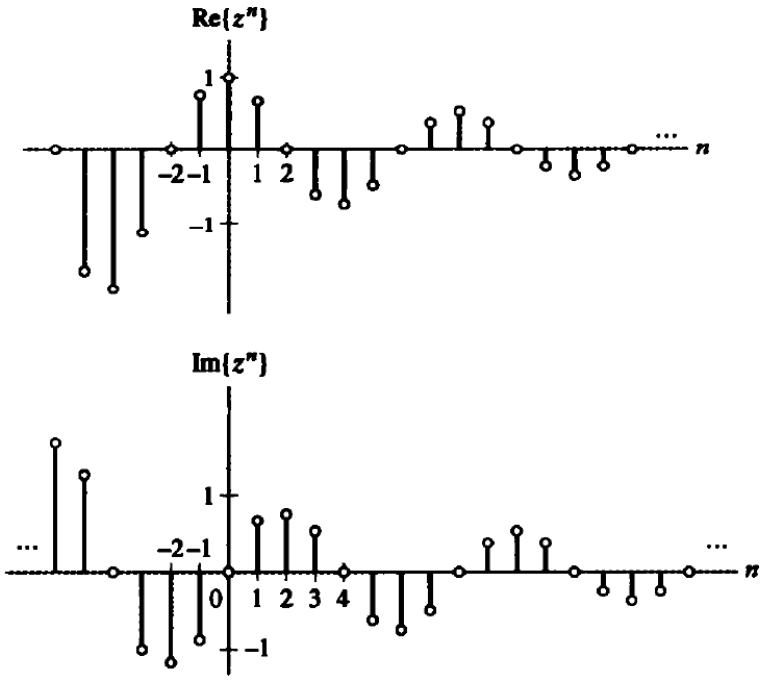


Figure 1: Real and imaginary parts of discrete-time signal z^n

If we define the transfer function as:

$$H(z) = \sum_{k=-\infty}^{\infty} h[k]\, z^{-k} \qquad (Eq. 4)$$

Then, we may write:

$$H\{z^n\} = H(z)z^n \qquad (Eq. 5)$$

If we express $H(z)$ in the poles form as $H(z) = |H(z)|e^{j\phi(z)}$, then the system output is written as:

$$y[n] = |H(z)|e^{j\phi(z)}z^n \qquad (Eq. 6)$$

More generally, the z-transform of an arbitrary signal $x[n]$ is:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]\, z^{-n} \qquad (Eq. 7)$$

The z-transform exists when the infinite sum of the equation above converges. A necessary condition for the convergence is absolute summability of $x[n]z^{-n}$. Since $|x[n]z^{-n}| = |x[n]r^{-n}|$, we must have

$$\sum_{n=-\infty}^{\infty} |x[n]r^{-n}| < \infty \qquad (Eq.8)$$

The range of r for which this condition is satisfied in termed the region of convergence (ROC) of the z-transform.

Practical Exercise 1 – Plots of z-Transform

You are required to do the analysis of discrete-time systems, attempt the following tasks.

a. For a given z-transform relationship, perform the following simulation tasks in MATLAB:
   i. Plot of $r^{-n}\cos(\omega n)$ when $r < 1$ and $\omega = \pi/4$ per samples. [10 marks]
   ii. Plot of $r^{-n}\cos(\omega n)$ when $r = 1$ and $\omega = \pi/4$ per samples. [10 marks]
   iii. Plot of $r^{-n}\cos(\omega n)$ when $r > 1$ and $\omega = \pi/4$ per samples. [10 marks]
b. Describe the results of the simulations as in part (a)i - (a)iii. [5 marks]

B. Forms of z-Transforms

There are two forms of z-transform e.g. forward and inverse z-transform. In this section, we will look into analysing their simulations and characteristics in MATLAB.

First of all, the z-transform of a discrete-time signal $x(n)$ could be defined as the power series.

$$X(z) = \sum_{n=-\infty}^{+\infty} x(n)z^{-n} \qquad (Eq.9)$$

Where:

- z is a complex variable.
- n is the order of function.

The relation above sometimes is called the direct z-transform because it transforms the time-domain signal $x(n)$ into its complex-plane representation $X(z)$.

$$z = re^{j\omega} \qquad (Eq.10)$$

Hence

$$z^{-n} = r^{-n}e^{-j\omega n}$$
$$= r^{-n}[\cos(\omega n) - j\sin(\omega n)\,] \qquad (Eq.11)$$

Where:

- 'r' is a real number.
- $e^{j\omega}$ is Euler's Number.
- ω is the angular frequency in radians per sample.

In MATLAB, we use `ztrans()` function to perform the z-transform. As shown below, the ztrans() function simulates the z-transform of function f.

```
ztrans(f)
```

Then, like the other transforms that we learnt so far in the course, inverse z-transform is the opposite of the direct transform of z-transform. For a given z-transform of a function, we could determine the original function in the time domain by applying the inverse z-transform.

In MATLAB, we use `iztans()` function to perform the inverse z-transform. In the code given below, the `iztrans(F)` function returns the inverse z-transform of F.

```
iztrans(F)
```

Practical Exercise 2 – Forms of z-Transform

Find the relevant transforms of the following functions using simulation in MATLAB.

a. z-transform of the functions given below.

i. $x_1(n) = a^n$ [5 marks]

ii. $x_2(n) = (1/4)^n$ [5 marks]

iii. $x_3(n) = \sin(\omega * n)$ [5 marks]

b. Inverse z-transform of the discrete sequence given below. [5 marks]

$$ix(n) = 2\left[\frac{z}{(2z-1)}\right]$$

C. Analysis of Discrete Systems with z-Transform

In z-transform, the z-plane and impulse response of a given discrete systems could be typically used to analyse the system's characteristics and behaviour.

It is convenient to represent the complex number z as a location in a complex plane termed the z-plane and depicted graphically in the figure below.

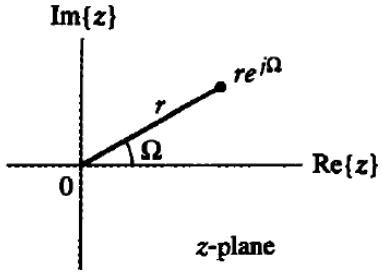


Figure 2: A point located at distance r from the origin and at angle Ω relative to real axis

The point $z = re^{j\Omega}$ is located at a distance r from the origin and an andle Ω from the positive real axis. Note that if $x[n]$ is absolutely summable, then the DTFT is obtained from the z-transform by setting $r = 1$, or substituting $z = e^{j\Omega}$ into Eq. 7, that is:

$$X(e^{j\Omega}) = X(z)|_{z=e^{j\Omega}} \qquad (Eq. 12)$$

The equation $z = e^{j\Omega}$ describes a circle of unit radius centred on the origin in the z-plane, as illustrated in the figure below.

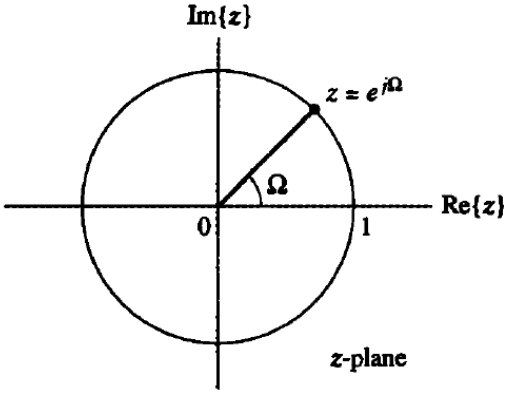


Figure 3: The unit circle, $z = e^{j\Omega}$, in the z-plane

The contour in the diagram is termed the unit circle in the z-plane. The frequency Ω in the DTFT corresponds to the point on the unit circle at an angle of Ω with respect to the positive real-axis. As the

discrete time frequency Ω goes from $-\pi$ to π, we take one trip around the unit circle. We say that the DTFT corresponds to the z-transform evaluated on the unit circle.

Practical Exercise 3 – Analysis of Discrete Systems with z-Transform

We can analyse the characteristics and behaviour of discrete system from the z-plane and impulse response. Plot the z-plane and impulse response of the following discrete systems in MATLAB.

a. System with complex conjugate poles. [10 marks]

$$H_1(z) = \frac{z}{z^2 - 0.707z + 0.2499}$$

Or

$$H_1(z) = \frac{z}{[z - (0.3535 + j0.3535)][z - (0.3535 - j0.3535)]}$$

b. System with complex conjugate poles in unit circle. [10 marks]

$$H_2(z) = \frac{z}{z^2 - 1.4144z + 1}$$

c. System with complex conjugate poles outside of unit circle. [10 marks]

$$H_3(z) = \frac{z^2}{z^2 - 2.02z + 1.14}$$

d. Describe the characteristics and behaviours of the given discrete systems from the z-plane and impulse response simulations in parts (a)-(c). Formulate stability criteria in z-transform. [10 marks]

C. Partial Fraction Expansion

There are two methods for determining the inverse z-transform e.g. partial fraction expansion and a method that expresses $X(z)$ as a power series in z^{-1}. In most cases, the partial fraction expansion is used for finding the inverse z-transform.

In the study of LTI systems, we often encounter z-transform that are a rational function of z^{-1}. Let

$$X(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}} \qquad (Eq. 13)$$

And assume that: $M < N$. If $M \geq N$, then we may use the long division to express $X(z)$ in the form:

$$X(z) = \sum_{k=0}^{M-N} f_k\, z^{-k} + \frac{B(z)}{A(z)} \qquad (Eq.14)$$

The numerator polynomial $B(z)$ now has order one less than that of denominator polynomial and the partial-fraction expansion is applied to determine the inverse transform of $B(z)/A(z)$. The inverse z-transform of the terms in the sum are obtained from the pair z-transform table and the time-shift property of the z-transform.

In MATLAB, for finding partial fraction in z-transform, we can use the following function:

```
[R,P,K]=residueZ(num,den)
```

Where: `R`, `P` and `K` forms the partial fraction in the following forms:

$$X(z) = \frac{R_1 z}{(z-p_1)} + \frac{R_2 z}{(z-p_2)} + \cdots + K$$

Practical Exercise 4 – Partial Fraction in z-Transform

Find the partial fraction of following discrete sequence expression: [5 marks]

$$X(z) = \frac{4 - (7/4)z^{-1} + (1/4)z^{-2}}{1 - (3/4)z^{-1} + (1/8)z^{-2}}$$

Appendix 1 - Marking Rubric

Student ID : _____

Student Name : _____

No	Description	Marks	Your Marks	Remarks
1ai	Plot of $r^{-n}\cos(\omega n)$ when $r < 1$ and ω = $\pi/4$ per samples.	10		
1aii	Plot of $r^{-n}\cos(\omega n)$ when $r = 1$ and ω = $\pi/4$ per samples.	10		
1aiii	Plot of $r^{-n}\cos(\omega n)$ when $r > 1$ and ω = $\pi/4$ per samples.	10		
1b	Description of simulations in part (a)i-(a)iii.	5		
2ai	z-transform of the function 1.	5		
2aii	z-transform of the function 2.	5		
2aiii	z-transform of the function 3.	5		
2b	Inverse z-transform of the given sequence.	5		
3a	Plots the z-plane and impulse response of system 1.	10		
3b	Plots the z-plane and impulse response of system 2.	10		
3c	Plots the z-plane and impulse response of system 3.	10		
3d	Characteristics and behaviours description of the systems from the z-plane and impulse response simulations in parts (a)-(c). Formulation of stability criteria.	10		
4	Partial fraction of the discrete sequence	5		
Total		100		

Feedback:

Appendix

Appendix 1 – Inverse z-Transform

ztrans

Z-transform

Syntax

```
ztrans(f)
ztrans(f,transVar)
ztrans(f,var,transVar)
```

Description

`ztrans(f)` finds the Z-Transform of f. By default, the independent variable is n and the transformation variable is z. If f does not contain n, ztrans uses symvar.

`ztrans(f,transVar)` uses the transformation variable transVar instead of z.

`ztrans(f,var,transVar)` uses the independent variable var and transformation variable transVar instead of n and z, respectively.

Examples

Z-Transform of Symbolic Expression

Compute the Z-transform of sin(n). By default, the transform is in terms of z.

```
syms n
f = sin(n);
ztrans(f)

ans =
(z*sin(1))/(z^2 - 2*cos(1)*z + 1)
```

Specify Independent Variable and Transformation Variable

Compute the Z-transform of exp(m+n). By default, the independent variable is n and the transformation variable is z.

```
syms m n
f = exp(m+n);
ztrans(f)

ans =
(z*exp(m))/(z - exp(1))
```

Specify the transformation variable as y. If you specify only one variable, that variable is the transformation variable. The independent variable is still n.

```
syms y
ztrans(f,y)

ans =
(y*exp(m))/(y - exp(1))
```

Specify both the independent and transformation variables as m and y in the second and third arguments, respectively.

```
ztrans(f,m,y)

ans =
(y*exp(n))/(y - exp(1))
```

Z-Transforms Involving Heaviside Function and Binomial Coefficient

Compute the Z-transform of the Heaviside function and the binomial coefficient.

```
syms n z
ztrans(heaviside(n-3),n,z)

ans =
(1/(z - 1) + 1/2)/z^3
ztrans(nchoosek(n,2))
ans =
z/(z - 1)^3
```

z-Transform of Array Inputs

Find the Z-transform of the matrix M. Specify the independent and transformation variables for each matrix entry by using matrices of the same size. When the arguments are non-scalars, ztrans acts on them element-wise.

```
syms a b c d w x y z
M = [exp(x) 1; sin(y) i*z];
vars = [w x; y z];
transVars = [a b; c d];
ztrans(M,vars,transVars)

ans =
[                (a*exp(x))/(a - 1),       b/(b - 1)]
[ (c*sin(1))/(c^2 - 2*cos(1)*c + 1), (d*1i)/(d - 1)^2]
```

If ztrans is called with both scalar and non-scalar arguments, then it expands the scalars to match the non-scalars by using scalar expansion. Non-scalar arguments must be the same size.

```
syms w x y z a b c d
ztrans(x,vars,transVars)
```

```
ans =
[ (a*x)/(a - 1),   b/(b - 1)^2]
[ (c*x)/(c - 1), (d*x)/(d - 1)]
```

Z-Transform of Symbolic Function

Compute the Z-transform of symbolic functions. When the first argument contains symbolic functions, then the second argument must be a scalar.

```
syms f1(x) f2(x) a b
f1(x) = exp(x);
f2(x) = x;
ztrans([f1 f2],x,[a b])

ans =
[ a/(a - exp(1)), b/(b - 1)^2]
```

Appendix 2 – Inverse z-Transform

iztrans

Inverse Z-transform

Syntax

```
iztrans(F)
iztrans(F,transVar)
iztrans(F,var,transVar)
```

Description

`iztrans(F)` returns the Inverse Z-Transform of F. By default, the independent variable is z and the transformation variable is n. If F does not contain z, iztrans uses the function symvar.

`iztrans(F,transVar)` uses the transformation variable transVar instead of n.

`iztrans(F,var,transVar)` uses the independent variable var and transformation variable transVar instead of z and n respectively.

Examples

Inverse Z-Transform of Symbolic Expression

Compute the inverse Z-transform of 2*z/(z-2)^2. By default, the inverse transform is in terms of n.

```
syms z
F = 2*z/(z-2)^2;
iztrans(F)

ans =
2^n + 2^n*(n - 1)
```

Specify Independent Variable and Transformation Variable

Compute the inverse Z-transform of 1/(a*z). By default, the independent and transformation variables are z and n, respectively.

```
syms z a
F = 1/(a*z);
iztrans(F)

ans =
kroneckerDelta(n - 1, 0)/a
```

Specify the transformation variable as m. If you specify only one variable, that variable is the transformation variable. The independent variable is still z.

```
syms m
```

```
iztrans(F,m)

ans =
kroneckerDelta(m - 1, 0)/a
```

Specify both the independent and transformation variables as a and m in the second and third arguments, respectively.

```
iztrans(F,a,m)

ans =
kroneckerDelta(m - 1, 0)/z
```

Inverse Z-Transforms Involving Kronecker Delta Function

Compute the inverse Z-transforms of these expressions. The results involve the Kronecker Delta function.

```
syms n z
iztrans(1/z,z,n)

ans =
kroneckerDelta(n - 1, 0)

f = (z^3 + 3*z^2)/z^5;
iztrans(f,z,n)

ans =
kroneckerDelta(n - 2, 0) + 3*kroneckerDelta(n - 3, 0)
```

Inverse Z-Transform of Array Inputs

Find the inverse Z-transform of the matrix M. Specify the independent and transformation variables for each matrix entry by using matrices of the same size. When the arguments are nonscalars, iztrans acts on them element-wise.

```
syms a b c d w x y z
M = [exp(x) 1; sin(y) i*z];
vars = [w x; y z];
transVars = [a b; c d];
iztrans(M,vars,transVars)

ans =
[ exp(x)*kroneckerDelta(a, 0), kroneckerDelta(b, 0)]
[       iztrans(sin(y), y, c),   iztrans(z, z, d)*1i]
```

If iztrans is called with both scalar and nonscalar arguments, then it expands the scalars to match the nonscalars by using scalar expansion. Nonscalar arguments must be the same size.

```
syms w x y z a b c d
iztrans(x,vars,transVars)

ans =
[ x*kroneckerDelta(a, 0),       iztrans(x, x, b)]
[ x*kroneckerDelta(c, 0), x*kroneckerDelta(d, 0)]
```

Inverse Z-Transform of Symbolic Function

Compute the Inverse Z-transform of symbolic functions. When the first argument contains symbolic functions, then the second argument must be a scalar.

```
syms f1(x) f2(x) a b
f1(x) = exp(x);
f2(x) = x;
```

```
iztrans([f1, f2],x,[a, b])

ans =
[ iztrans(exp(x), x, a), iztrans(x, x, b)]
```

If Inverse Z-Transform Cannot Be Found

If iztrans cannot compute the inverse transform, it returns an unevaluated call.

```
syms F(z) n
F(z) = exp(z);
f = iztrans(F,z,n)
f = iztrans(exp(z), z, n)
```

Return the original expression by using ztrans.

```
ztrans(f,n,z)

ans =
exp(z)
```

Appendix 3 – Partial Fraction Expansion

residuez

Z-transform partial-fraction expansion

Syntax

```
[ro,po,ko] = residuez(bi,ai)
[bo,ao] = residuez(ri,pi,ki)
```

Description

`[ro,po,ko] = residuez(bi,ai)` finds the residues, poles, and direct terms of a partial fraction expansion of the ratio of numerator and denominator polynomials, b and a.

`[bo,ao] = residuez(ri,pi,ki)` with three input arguments and two output arguments, converts the partial fraction expansion back to polynomials with coefficients in row vectors b and a.

Examples

Partial-Fraction Expansion of IIR Lowpass Filter

Compute the partial-fraction expansion corresponding to the third-order IIR lowpass filter described by the transfer function

$$H(z)=\frac{0.05634(1+z^{-1})(1-1.0166z^{-1}+z^{-2})}{(1-0.683z^{-1})(1-1.4461z^{-1}+0.7957z^{-2})}.$$

Express the numerator and denominator as polynomial convolutions.

```
b0 = 0.05634;
b1 = [1  1];
b2 = [1 -1.0166 1];
a1 = [1 -0.683];
a2 = [1 -1.4461 0.7957];

b = b0*conv(b1,b2);
a = conv(a1,a2);
```

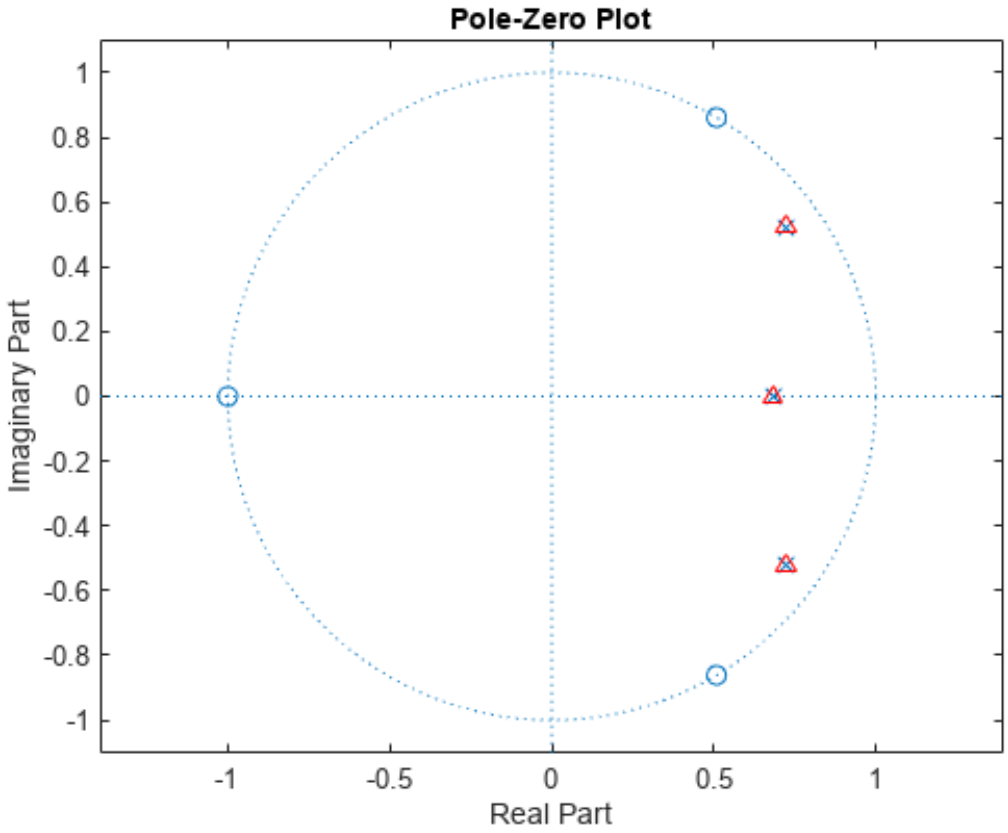
Compute the residues, poles, and direct terms of the partial-fraction expansion.

```
[r,p,k] = residuez(b,a)
r = 3×1 complex

  -0.1153 - 0.0182i
  -0.1153 + 0.0182i
```

```
   0.3905 + 0.0000i

p = 3×1 complex

   0.7230 + 0.5224i
   0.7230 - 0.5224i
   0.6830 + 0.0000i

k = -0.1037
```

Plot the poles and zeros of the transfer function and overlay the poles you just found.

```
zplane(b,a)
hold on
plot(p,'^r')
hold off
```



Use residuez again to reconstruct the transfer function.

[bn,an] = residuez(r,p,k)

bn = *1×4*

0.0563 -0.0009 -0.0009 0.0563

an = *1×4*

1.0000 -2.1291 1.7834 -0.5435