KALMAN FILTER GENERALIZATIONS

5.1: Maintaining symmetry of covariance matrices

- The Kalman filter as described so far is theoretically correct, but has known vulnerabilities and limitations in practical implementations.
- In this unit of notes, we consider the following issues:
 - 1. Improving numeric robustness;
 - 2. Sequential measurement processing and square-root filtering;
 - 3. Dealing with auto- and cross-correlated sensor or process noise;
 - 4. Extending the filter to prediction and smoothing;
 - 5. Reduced-order filtering;
 - 6. Using residue analysis to detect sensor faults.

Improving numeric robustness

- Within the filter, the covariance matrices $\Sigma_{\tilde{x},k}^{-}$ and $\Sigma_{\tilde{x},k}^{+}$ must remain
 - 1. Symmetric, and
 - 2. Positive definite (all eigenvalues strictly positive).
- It is possible for both conditions to be violated due to round-off errors in a computer implementation.
- We wish to find ways to limit or eliminate these problems.

Dealing with loss of symmetry

- The cause of covariance matrices becoming asymmetric or non-positive definite must be due to either the time-update or measurement-update equations of the filter.
- Consider first the time-update equation:

$$\Sigma_{\tilde{x},k}^{-} = A \Sigma_{\tilde{x},k-1}^{+} A^{T} + \Sigma_{\tilde{w}}.$$

- Because we are adding two positive-definite quantities together, the result must be positive definite.
- A "suitable implementation" of the products of the matrices will avoid loss of symmetry in the final result.
- Consider next the measurement-update equation:

$$\Sigma_{\tilde{x},k}^{+} = \Sigma_{\tilde{x},k}^{-} - L_k C_k \Sigma_{\tilde{x},k}^{-}.$$

- Theoretically, the result is positive definite, but due to the subtraction operation it is possible for round-off errors in an implementation to result in a non-positive-definite solution.
- The problem may be mitigated in part by computing instead

$$\Sigma_{\tilde{x},k}^+ = \Sigma_{\tilde{x},k}^- - L_k \Sigma_{\tilde{z},k} L_k^T.$$

• This may be proven correct via

$$\Sigma_{\tilde{x},k}^{+} = \Sigma_{\tilde{x},k}^{-} - L_k \Sigma_{\tilde{z},k} L_k^T$$

$$= \Sigma_{\tilde{x},k}^{-} - L_k \Sigma_{\tilde{z},k} \left(\Sigma_{\tilde{x},k}^{-} C_k^T \Sigma_{\tilde{z},k}^{-1} \right)^T$$

$$= \Sigma_{\tilde{x},k}^{-} - L_k \Sigma_{\tilde{z},k} \Sigma_{\tilde{z},k}^{-1} C_k \Sigma_{\tilde{x},k}^{-}$$

$$= \Sigma_{\tilde{x},k}^{-} - L_k C_k \Sigma_{\tilde{x},k}^{-}.$$

- With a "suitable implementation" of the products in the $L_k \Sigma_{\tilde{z},k} L_k^T$ term, symmetry can be guaranteed. However, the subtraction may still give a non-positive definite result if there is round-off error.
- A better solution is the Joseph form covariance update.

$$\Sigma_{\tilde{x},k}^{+} = [I - L_k C_k] \Sigma_{\tilde{x},k}^{-} [I - L_k C_k]^T + L_k \Sigma_{\tilde{v}} L_k^T.$$

• This may be proven correct via

$$\begin{split} \Sigma_{\tilde{x},k}^{+} &= [I - L_k C_k] \, \Sigma_{\tilde{x},k}^{-} [I - L_k C_k]^T + L_k \Sigma_{\tilde{v}} L_k^T \\ &= \Sigma_{\tilde{x},k}^{-} - L_k C_k \Sigma_{\tilde{x},k}^{-} - \Sigma_{\tilde{x},k}^{-} C_k^T L_k^T + L_k C_k \Sigma_{\tilde{x},k}^{-} C_k^T L_k^T + L_k \Sigma_{\tilde{v}} L_k^T \\ &= \Sigma_{\tilde{x},k}^{-} - L_k C_k \Sigma_{\tilde{x},k}^{-} - \Sigma_{\tilde{x},k}^{-} C_k^T L_k^T + L_k \left(C_k \Sigma_{\tilde{x},k}^{-} C_k^T + \Sigma_{\tilde{v}} \right) L_k^T \\ &= \Sigma_{\tilde{x},k}^{-} - L_k C_k \Sigma_{\tilde{x},k}^{-} - \Sigma_{\tilde{x},k}^{-} C_k^T L_k^T + L_k \Sigma_{\tilde{z},k} L^T \\ &= \Sigma_{\tilde{x},k}^{-} - L_k C_k \Sigma_{\tilde{x},k}^{-} - \Sigma_{\tilde{x},k}^{-} C_k^T L_k^T + \left(\Sigma_{\tilde{x},k}^{-} C_k^T \Sigma_{\tilde{z},k}^{-1} \right) \Sigma_{\tilde{z},k} L^T \\ &= \Sigma_{\tilde{x},k}^{-} - L_k C_k \Sigma_{\tilde{x},k}^{-} . \end{split}$$

- Because the subtraction occurs in the "squared" term, this form "guarantees" a positive definite result.
- If we end up with a negative definite matrix (numerics), we can replace it by the nearest symmetric positive semidefinite matrix.¹
- Omitting the details, the procedure is:
 - Calculate singular-value decomposition: $\Sigma = USV^T$.
 - Compute $H = VSV^T$.
 - Replace Σ with $(\Sigma + \Sigma^T + H + H^T)/4$.
- ¹ Nicholas J. Higham, "Computing a Nearest Symmetric Positive Semidefinite Matrix," *Linear Algebra and its Applications*, vol. 103, pp. 103–118, 1988.

5.2: Sequential processing of measurements

- There are still improvements that may be made. We can:
 - Reduce the computational requirements of the Joseph form,
 - Increase the precision of the numeric accuracy.
- One of the computationally intensive operations in the Kalman filter is the matrix inverse operation in $L_k = \sum_{\tilde{x},k}^{-} C_k^T \sum_{\tilde{z},k}^{-1}$.
- Using matrix inversion via Gaussian elimination (the most straightforward approach), is an O(m³) operation, where m is the dimension of the measurement vector.
- If there is a single sensor, this matrix inverse becomes a scalar division, which is an O(1) operation.
- Therefore, if we can break the *m* measurements into *m* single-sensor measurements and update the Kalman filter that way, there is opportunity for significant computational savings.

Sequentially processing independent measurements

 We start by assuming that the sensor measurements are independent. That is, that

$$\Sigma_{\tilde{v}} = \operatorname{diag}\left[\sigma_{\tilde{v}_1}^2, \cdots \sigma_{\tilde{v}_m}^2 \right]$$

- We will use colon ":" notation to refer to the measurement number.
 For example, *z_{k:1}* is the measurement from sensor 1 at time *k*.
- Then, the measurement is

$$z_{k} = \begin{bmatrix} z_{k:1} \\ \vdots \\ z_{k:m} \end{bmatrix} = C_{k}x_{k} + v_{k} = \begin{bmatrix} C_{k:1}^{T}x_{k} + v_{k:1} \\ \vdots \\ C_{k:m}^{T}x_{k} + v_{k:m} \end{bmatrix}$$

,

where $C_{k:1}^T$ is the first row of C_k (for example), and $v_{k:1}$ is the sensor noise of the first sensor at time k, for example.

- We will consider this a sequence of scalar measurements $z_{k:1} \dots z_{k:m}$, and update the state estimate and covariance estimates in *m* steps.
- We initialize the measurement update process with $\hat{x}_{k:0}^+ = \hat{x}_k^-$ and $\Sigma_{\tilde{x},k:0}^+ = \Sigma_{\tilde{x},k}^-$.
- Consider the measurement update for the *i*th measurement, $z_{k:i}$

$$\hat{x}_{k:i}^{+} = \mathbb{E}[x_{k} \mid \mathbb{Z}_{k-1}, z_{k:1} \dots z_{k:i}]$$

$$= \mathbb{E}[x_{k} \mid \mathbb{Z}_{k-1}, z_{k:1} \dots z_{k:i-1}] + L_{k:i}(z_{k:i} - \mathbb{E}[z_{k} \mid \mathbb{Z}_{k-1}, z_{k:1} \dots z_{k:i-1}])$$

$$= \hat{x}_{k:i-1}^{+} + L_{k:i}(z_{k:i} - C_{k:i}^{T} \hat{x}_{k:i-1}^{+}).$$

Generalizing from before

$$L_{k:i} = \mathbb{E}[\tilde{x}_{k:i-1}^+ \tilde{z}_{k:i}^T] \Sigma_{\tilde{z}_{k:i}}^{-1}.$$

 Next, we recognize that the variance of the innovation corresponding to measurement z_{k:i} is

$$\Sigma_{\tilde{z}_{k:i}} = \sigma_{\tilde{z}_{k:i}}^2 = C_{k:i}^T \Sigma_{\tilde{x},k:i-1}^+ C_{k:i} + \sigma_{\tilde{v}_i}^2.$$

• The corresponding gain is $L_{k:i} = \frac{\sum_{\tilde{x},k:i-1}^{+}C_{k:i}}{\sigma_{\tilde{z}_{k:i}}^{2}}$ and the updated state is $\hat{x}_{k:i}^{+} = \hat{x}_{k:i-1}^{+} + L_{k:i} \left[z_{k:i} - C_{k:i}^{T} \hat{x}_{k:i-1}^{+} \right]$

with covariance

$$\Sigma_{\tilde{x},k:i}^{+} = \Sigma_{\tilde{x},k:i-1}^{+} - L_{k:i}C_{k:i}^{T}\Sigma_{\tilde{x},k:i-1}^{+}.$$

The covariance update can be implemented as

$$\Sigma_{\tilde{x},k:i}^{+} = \Sigma_{\tilde{x},k:i-1}^{+} - \frac{\Sigma_{\tilde{x},k:i-1}^{+}C_{k:i}C_{k:i}^{T}\Sigma_{\tilde{x},k:i-1}^{+}}{C_{k:i}^{T}\Sigma_{\tilde{x},k:i-1}^{+}C_{k:i} + \sigma_{\tilde{v}_{i}}^{2}}.$$

An alternative update is the Joseph form,

$$\Sigma_{\tilde{x},k:i}^{+} = \left[I - L_{k:i}C_{k:i}^{T}\right]\Sigma_{\tilde{x},k:i}^{+}\left[I - L_{k:i}C_{k:i}^{T}\right]^{T} + L_{k:i}\sigma_{\tilde{v}_{i}}^{2}L_{k:i}^{T}.$$

• The final measurement update gives $\hat{x}_{k}^{+} = \hat{x}_{k:m}^{+}$ and $\Sigma_{\tilde{x},k}^{+} = \Sigma_{\tilde{x},k:m}^{+}$.

Sequentially processing correlated measurements

- The above process must be modified to accommodate the situation where sensor noise is correlated among the measurements.
- Assume that we can factor the matrix $\Sigma_{\tilde{v}} = S_v S_v^T$, where S_v is a lower-triangular matrix (for symmetric positive-definite $\Sigma_{\tilde{v}}$, we can).
 - The factor S_v is a kind of a matrix square root, and will be important in a number of places in this course.
 - It is known as the "Cholesky" factor of the original matrix.
 - In MATLAB, Sv = chol(SigmaV, 'lower');
 - Be careful: MATLAB's default answer (without specifying "lower") is an upper-triangular matrix, which is not what we're after.
- The Cholesky factor has strictly positive elements on its diagonal (positive eigenvalues), so is guaranteed to be invertible.
- Consider a modification to the output equation of a system having correlated measurements

$$z_k = Cx_k + v_k$$
$$\bar{z}_k = \mathcal{S}_v^{-1} z_k = \mathcal{S}_v^{-1} Cx_k + \mathcal{S}_v^{-1} v_k$$
$$\bar{z}_k = \bar{C} x_k + \bar{v}_k.$$

• Note that we will use the "bar" decoration ($\overline{\cdot}$) frequently in this chapter of notes.

- It rarely (if ever) indicates the mean of that quantity.
- Rather, it refers to a definition having similar meaning to the original symbol.
- For example, \bar{z}_k is a (computed) output value, similar in interpretation to the measured output value z_k .
- Consider now the covariance of the modified noise input $\bar{v}_k = S_v^{-1} v_k$

$$\Sigma_{\tilde{\tilde{v}}_k} = \mathbb{E}[\bar{v}_k \bar{v}_k^T]$$
$$= \mathbb{E}[\mathcal{S}_v^{-1} v_k v_k^T \mathcal{S}_v^{-T}]$$
$$= \mathcal{S}_v^{-1} \Sigma_{\tilde{v}} \mathcal{S}_v^{-T} = I.$$

- Therefore, we have identified a transformation that de-correlates (and normalizes) measurement noise.
- Using this revised output equation, we use the prior method.
- We start the measurement update process with $\hat{x}_{k:0}^+ = \hat{x}_k^-$ and $\Sigma_{\tilde{x},k:0}^+ = \Sigma_{\tilde{x},k}^-$.
- The Kalman gain is $\bar{L}_{k:i} = \frac{\sum_{\tilde{x},k:i-1}^{+} \bar{C}_{k:i}}{\bar{C}_{k:i}^{T} \sum_{\tilde{x},k:i-1}^{+} \bar{C}_{k:i} + 1}$ and the updated state is $\hat{x}_{k:i}^{+} = \hat{x}_{k:i-1}^{+} + \bar{L}_{k:i} \left[\bar{z}_{k:i} - \bar{C}_{k:i}^{T} \hat{x}_{k:i-1}^{+} \right]$ $= \hat{x}_{k:i-1}^{+} + \bar{L}_{k:i} \left[(S_{p}^{-1} z_{k})_{i} - \bar{C}_{k:i}^{T} \hat{x}_{k:i-1}^{+} \right].$

with covariance

$$\Sigma_{\tilde{x},k:i}^{+} = \Sigma_{\tilde{x},k:i-1}^{+} - \bar{L}_{k:i}\bar{C}_{k:i}^{T}\Sigma_{\tilde{x},k:i-1}^{+}$$

(which may also be computed with a Joseph form update, for example).

• The final measurement update gives $\hat{x}_{k}^{+} = \hat{x}_{k:m}^{+}$ and $\Sigma_{\tilde{x}:k}^{+} = \Sigma_{\tilde{x},k:m}^{+}$.

LDL updates for correlated measurements

 An alternative to the Cholesky decomposition for factoring the covariance matrix is the LDL decomposition

$$\Sigma_{\tilde{v}} = \mathcal{L}_{v} \mathcal{D}_{v} \mathcal{L}_{v}^{T}$$
,

where \mathcal{L}_v is lower-triangular and \mathcal{D}_v is diagonal (with positive entries).

- In MATLAB, [L,D] = ldl(SigmaV);
- The Cholesky decomposition is related to the LDL decomposition via

$$\mathcal{S}_v = \mathcal{L}_v \mathcal{D}_v^{1/2}.$$

- Both texts show how to use the LDL decomposition to perform a sequential measurement update.
- A computational advantage of LDL over Cholesky is that no square-root operations need be taken. (We can avoid finding D_v^{1/2}.)
- A pedagogical advantage of introducing the Colesky decomposition is that we use it later on.

5.3: Square-root filtering

- The modifications to the basic Kalman filter that we have described so far are able to
 - Ensure symmetric, positive-definite covariance matrices;
 - Speed up the operation of a multiple-measurement Kalman filter.
- The filter is still sensitive to finite word length: no longer in the sense of causing divergence, but in the sense of not converging to as good a solution as possible.
- Consider the set of numbers: 1,000,000; 100; 1. There are six orders of magnitude in the spread between the largest and smallest.
- Now consider a second set of numbers: 1,000; 10; 1. There are only three orders of magnitude in spread.
- But, the second set is the square root of the first set: We can reduce dynamic range (number of bits required to implement a given precision of solution) by using square roots of numbers.
- For example, we can get away with single-precision math instead of double-precision math.
- The place this really shows up is in the eigenvalue spread of covariance matrices. If we can use square-root matrices instead, that would be better.
- Consider the Cholesky factorization from before. Define $\Sigma_{\tilde{x},k}^+ = S_{\tilde{x},k}^+ \left(S_{\tilde{x},k}^+\right)^T$ and $\Sigma_{\tilde{x},k}^- = S_{\tilde{x},k}^- \left(S_{\tilde{x},k}^-\right)^T$.

- We would like to be able to compute the covariance time updates and measurement updates in terms of S[±]_{x,k} instead of Σ[±]_{x,k}. Let's take the steps in order.
- SR-KF step 1a: State estimate time update.
 - We compute

$$\hat{x}_k^- = A_{k-1}\hat{x}_{k-1}^+ + B_{k-1}u_{k-1}.$$

No change in this step from standard KF.

SR-KF step 1b: Error covariance time update.

We start with standard step

$$\Sigma_{\tilde{x},k}^{-} = A_{k-1} \Sigma_{\tilde{x},k-1}^{+} A_{k-1}^{T} + \Sigma_{\tilde{w}}.$$

• We would like to write this in terms of Cholesky factors

$$\mathcal{S}_{\tilde{x},k}^{-}\left(\mathcal{S}_{\tilde{x},k}^{-}\right)^{T} = A_{k-1}\mathcal{S}_{\tilde{x},k-1}^{+}\left(\mathcal{S}_{\tilde{x},k-1}^{+}\right)^{T}A_{k-1}^{T} + \mathcal{S}_{\tilde{w}}\mathcal{S}_{\tilde{w}}^{T}.$$

- One option is to compute the right side, then take the Cholesky decomposition to compute the factors on the left side. This is computationally too intensive.
- Instead, start by noticing that we can write the equation as

$$\mathcal{S}_{\tilde{x},k}^{-}\left(\mathcal{S}_{\tilde{x},k}^{-}\right)^{T} = \left[\begin{array}{cc} A_{k-1}\mathcal{S}_{\tilde{x},k-1}^{+}, & \mathcal{S}_{\tilde{w}} \end{array}\right] \left[\begin{array}{cc} A_{k-1}\mathcal{S}_{\tilde{x},k-1}^{+}, & \mathcal{S}_{\tilde{w}} \end{array}\right]^{T} \\ = MM^{T}.$$

- This might at first appear to be exactly what we desire, but the problem is that $S^{-}_{\tilde{x},k}$ is and $n \times n$ matrix, whereas *M* is an $n \times 2n$ matrix.
- But, it is at least a step in the right direction. Enter the QR decomposition.

- *QR* decomposition: The QR decomposition algorithm computes two factors $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{n \times m}$ for a matrix $Z \in \mathbb{R}^{n \times m}$ such that Z = QR, Q is orthogonal, R is upper-triangular, and $m \ge n$.
 - The property of the QR factorization that is important here is that R is related to the Cholesky factor we wish to find.
 - Specifically, if $\tilde{R} \in \mathbb{R}^{n \times n}$ is the upper-triangular portion of R, then \tilde{R}^T is the Cholesky factor of $\Sigma = M^T M$.
 - That is, if $\tilde{R} = qr(M^T)^T$, where $qr(\cdot)$ performs the QR decomposition and returns the upper-triangular portion of *R* only, then \tilde{R} is the lower-triangular Cholesky factor of MM^T .
 - Continuing with our derivation, notice that if we form M as above, then compute \tilde{R} , we have our desired result.

$$\mathcal{S}_{\tilde{x},k}^{-} = \operatorname{qr}\left(\left[A_{k-1}\mathcal{S}_{\tilde{x},k-1}^{+}, \mathcal{S}_{\tilde{w}}\right]^{T}\right)^{T}$$

- The computational complexity of the QR decomposition is $\mathcal{O}(mn^2)$, whereas the complexity of the Cholesky factor is $\mathcal{O}(n^3/6)$ plus $\mathcal{O}(mn^2)$ to first compute MM^T .
- In MATLAB:

Sminus = qr([A*Splus,Sw]')';
Sminus = tril(Sminus(1:nx,1:nx));

SR-KF step 1c: Estimate system output.

As before, we estimate the system output as

$$\hat{z}_k = C_k \hat{x}_k^- + D_k u_k.$$

SR-KF step 2a: Estimator (Kalman) gain matrix.

- In this step, we must compute $L_k = \Sigma_{\tilde{x}\tilde{z},k}^{-} (\Sigma_{\tilde{z},k})^{-1}$.
- Recall that $\Sigma_{\tilde{x}\tilde{z},k}^{-} = \Sigma_{\tilde{x},k}^{-}C_{k}^{T}$ and $\Sigma_{\tilde{z},k} = C_{k}\Sigma_{\tilde{x},k}^{-}C_{k}^{T} + \Sigma_{\tilde{v}}$.
- We may find S_{*ž*,*k*} using the QR decomposition, as before. And, we already know S⁻_{*x*,*k*}.
- So, we can now write $L_k(S_{\tilde{z},k}S_{\tilde{z},k}^T) = \Sigma_{\tilde{x}\tilde{z},k}^-$.
- If z_k is not a scalar, this equation may often be computed most efficiently via back-substitution in two steps.
 - First, $(M)S_{\tilde{z},k}^T = \Sigma_{\tilde{x}\tilde{z},k}^-$ is found, and
 - Then $L_k S_{\tilde{z},k} = M$ is solved.
 - Back-substitution has complexity $\mathcal{O}(n^2/2)$.
 - Since $S_{\tilde{z},k}$ is already triangular, no matrix inversion need be done.
- Note that multiplying out $\Sigma_{\tilde{x},k}^{-} = S_{\tilde{x},k}^{-} \left(S_{\tilde{x},k}^{-}\right)^{T}$ in the computation of $\Sigma_{\tilde{x}\tilde{z},k}^{-}$ may drop some precision in L_k .
- However, this is not the critical issue.
- The critical issue is keeping $S_{\tilde{x},k}^{\pm}$ accurate for whatever L_k is used, which is something that we do manage to accomplish.
- In MATLAB:

Sz = qr([C*Sminus,Sv]')'; Sz = tril(Sz(1:nz,1:nz)); L = (Sminus*Sminus')*C'/Sz'/Sz;

SR-KF step 2b: State estimate measurement update.

This is done just as in the standard Kalman filter,

$$\hat{x}_k^+ = \hat{x}_k^- + L_k(z_k - \hat{z}_k).$$

SR-KF step 2c: Error covariance measurement update.

Finally, we update the error covariance matrix.

$$\Sigma_{\tilde{x},k}^{+} = \Sigma_{\tilde{x},k}^{-} - L_k \Sigma_{\tilde{z},k} L_k^T,$$

which can be written as,

$$\mathcal{S}_{\tilde{x},k}^{+}\left(\mathcal{S}_{\tilde{x},k}^{+}\right)^{T}=\mathcal{S}_{\tilde{x},k}^{-}\left(\mathcal{S}_{\tilde{x},k}^{-}\right)^{T}-L_{k}\mathcal{S}_{\tilde{z}}\mathcal{S}_{\tilde{z}}^{T}L_{k}^{T}.$$

- Note that the "-" sign prohibits us using the QR decomposition to solve this problem as we did before.
- Instead, we rely on the "Cholesky downdating" procedure.

In MATLAB,

```
% deal with MATLAB wanting upper-triangular Cholesky factor
Sx_ = Sminus';
% Want SigmaPlus = SigmaMinus - L*Sigmaz*L';
cov_update_vectors = L*Sz;
for j=1:length(zhat),
    Sx_ = cholupdate(Sx_,cov_update_vectors(:,j),'-');
end
% Re-transpose to undo MATLAB's strange Cholesky factor
Splus = Sx_';
```

 If you need to implement this kind of filter in a language other than MATLAB, a really excellent discussion of finding Cholesky factors, QR factorizations, and both Cholesky updating and downdating may be found in: G.W. Stewart, <u>Matrix Algorithms, Volume I: Basic</u> <u>Decompositions</u>, Siam, 1998. Pseudo-code is included.

Summary of the square-root linear Kalman filter

Linear state-space model: $\begin{aligned} x_k &= A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1} \\ z_k &= C_k x_k + D_k u_k + v_k, \end{aligned}$

where w_k and v_k are independent, zero-mean, Gaussian noise processes of covariance matrices $\Sigma_{\tilde{w}}$ and $\Sigma_{\tilde{v}}$, respectively.

Initialization: For k = 0, set

$$\begin{aligned} \hat{x}_0^+ &= \mathbb{E}[x_0] & \mathcal{S}_{\widetilde{w}} = \operatorname{chol}(\Sigma_{\widetilde{w}}, \text{'lower'}). \\ \Sigma_{\widetilde{x},0}^+ &= \mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]. & \mathcal{S}_{\widetilde{v}} = \operatorname{chol}(\Sigma_{\widetilde{v}}, \text{'lower'}). \\ \mathcal{S}_{\widetilde{x},0}^+ &= \operatorname{chol}(\Sigma_{\widetilde{x},0}^+, \text{'lower'}). \end{aligned}$$

Computation: For k = 1, 2, ... compute:

 $\begin{array}{ll} \mbox{State estimate time update:} & \hat{x}_k^- = A_{k-1} \hat{x}_{k-1}^+ + B_{k-1} u_{k-1}. \\ \mbox{Error covariance time update:} & \mathcal{S}_{\tilde{x},k}^- = \mbox{cholupdate} \left(\left(A_{k-1} \mathcal{S}_{\tilde{x},k-1}^+ \right)^T, \mathcal{S}_{\tilde{w}}^T \right)^T. \\ \mbox{Output estimate:} & \hat{z}_k = C_k \hat{x}_k^- + D_k u_k. \\ \mbox{Estimator gain matrix.}^* & \mathcal{S}_{\tilde{z},k} = \mbox{cholupdate} \left(\left(C_k \mathcal{S}_{\tilde{x},k}^- \right)^T, \mathcal{S}_{\tilde{v}}^T \right)^T. \\ \mbox{MS}_{\tilde{z},k}^T = \mathcal{S}_{\tilde{x},k}^- \left(\mathcal{S}_{\tilde{x},k}^- \right)^T \mathcal{C}_k^T, (\mbox{solved by backsubstitution}) \\ \mbox{L}_k \mathcal{S}_{\tilde{z},k} = M, (\mbox{solved by backsubstitution}) \\ \mbox{State estimate meas. update:} & \hat{x}_k^+ = \hat{x}_k^- + L_k (z_k - \hat{z}_k). \\ \mbox{Error covariance meas. update:} & \mathcal{S}_{\tilde{x},k}^+ = \mbox{cholupdate} \left(\left(\mathcal{S}_{\tilde{x},k}^- \right)^T, \left(L_k \mathcal{S}_{\tilde{z},k} \right)^T, '-' \right)^T. \end{array}$

*If a measurement is missed for some reason, then simply skip the measurement update for that iteration. That is, $L_k = 0$ and $\hat{x}_k^+ = \hat{x}_k^-$ and $\mathcal{S}_{\tilde{x},k}^+ = \mathcal{S}_{\tilde{x},k}^-$.

5.4: MATLAB code for the square-root Kalman filter steps

Coding a square-root Kalman filter in MATLAB is straightforward.

```
% Initialize simulation variables
SRSigmaW = chol(1, 'lower'); % Square-root process noise covar
SRSigmaV = chol(1,'lower'); % Square-root sensor noise covar
A = 1; B = 1; C = 1; D = 0; % Plant definition matrices
maxIter = 40;
xtrue = 0; xhat = 0 % Initialize true and estimated system initial state
SigmaX = 0.1; % Initialize Kalman filter covariance
SRSigmaX = chol(SigmaX, 'lower');
        % Unknown initial driving input: assume zero
u = 0;
% Reserve storage for variables we might want to plot/evaluate
xstore = zeros(maxIter+1, length(xtrue)); xstore(1,:) = xtrue;
xhatstore = zeros(maxIter, length(xhat));
SigmaXstore = zeros(maxIter, length(xhat)); % store diagonal only
for k = 1:maxIter,
  % SR-KF Step 1a: State estimate time update
  xhat = A*xhat + B*u; % use prior value of "u"
  % SR-KF Step 1b: Error covariance time update
  SRSigmaX = qr([A*SRSigmaX, SRSigmaW]')';
  SRSigmaX = tril(SRSigmaX(1:length(xhat),1:length(xhat)));
  % [Implied operation of system in background, with
  % input signal u, and output signal z]
  u = 0.5*randn(1) + cos(k/pi); % for example... (measured)
  w = SRSigmaW*randn(length(xtrue));
  v = SRSigmaV*randn(length(C*xtrue));
  ztrue = C*xtrue + D*u + v;  % y is based on present x and u
  xtrue = A*xtrue + B*u + w; % future x is based on present u
  % SR-KF Step 1c: Estimate system output
  zhat = C*xhat + D*u;
  % SR-KF Step 2a: Compute Kalman gain matrix
  % Note: "help mrdivide" to see how "division" is implemented
  SRSigmaZ = qr([C*SRSigmaX, SRSigmaV]')';
  SRSigmaZ = tril(SRSigmaZ(1:length(zhat),1:length(zhat)));
```

```
L = (SRSigmaX*SRSigmaX') *C'/SRSigmaZ'/SRSigmaZ;
  % SR-KF Step 2b: State estimate measurement update
  xhat = xhat + L*(ztrue - zhat);
  % SR-KF Step 2c: Error covariance measurement update
  Sx_ = SRSigmaX';
  cov_update_vectors = L*SRSigmaZ;
  for j=1:length(zhat),
    Sx_ = cholupdate(Sx_, cov_update_vectors(:, j), '-');
  end
  SRSigmaX = Sx_';
  % [Store information for evaluation/plotting purposes]
  xstore(k+1,:) = xtrue; xhatstore(k,:) = xhat;
  SigmaXstore(k,:) = diag(SRSigmaX*SRSigmaX');
end;
figure(1); clf;
plot(0:maxIter-1,xstore(1:maxIter),'k-',0:maxIter-1,xhatstore,'b--', ...
  0:maxIter-1, xhatstore+3*sqrt(SigmaXstore), 'm-.',...
  0:maxIter-1, xhatstore-3*sqrt(SigmaXstore), 'm-.'); grid;
title('Kalman filter in action'); xlabel('Iteration');
ylabel('State'); legend('true', 'estimate', 'bounds');
figure(2); clf;
plot(0:maxIter-1,xstore(1:maxIter)-xhatstore,'b-', ...
  0:maxIter-1, 3*sqrt (SigmaXstore), 'm--',...
  0:maxIter-1,-3*sqrt(SigmaXstore),'m--');
grid; legend('Error', 'bounds', 0); title('Error with bounds');
xlabel('Iteration'); ylabel('Estimation Error');
```



5.5: Cross-correlated process and measurement noises: Coincident

- The standard KF assumes that $\mathbb{E}[w_k v_j^T] = 0$. But, sometimes we may encounter systems where this is not the case.
- This might happen if both the physical process and the measurement system are affected by the same source of disturbance. Examples are changes of temperature, or inductive electrical interference.
- In this section, we assume that $\mathbb{E}[w_k w_j^T] = \Sigma_{\widetilde{w}} \delta_{kj}$, $\mathbb{E}[v_k v_j^T] = \Sigma_{\widetilde{v}} \delta_{kj}$, and $\mathbb{E}[w_k v_j^T] = \Sigma_{\widetilde{w}\widetilde{v}} \delta_{kj}$.
- Note that the correlation between noises is memoryless: the only correlation is at identical time instants.
- We can handle this case if we re-write the plant equation so that it has a new process noise that is uncorrelated with the measurement noise.
- Using an arbitrary matrix T (to be determined), we can write

$$x_{k+1} = A_k x_k + B_k u_k + w_k + T (z_k - C_k x_k - D_k u_k - v_k)$$

$$= (A_k - TC_k)x_k + (B_k - TD_k)u_k + w_k - Tv_k + Tz_k.$$

- Denote the new transition matrix $\bar{A}_k = A_k TC_k$, new input matrix as $\bar{B}_k = B_k TD_k$, and the new process noise as $\bar{w}_k = w_k Tv_k$.
- Further, denote the known (measured/computed) sequence as a new input $\bar{u}_k = T z_k$.
- Then, we can write a modified state space system

$$x_{k+1} = \bar{A}_k x_k + \bar{B}_k u_k + \bar{u}_k + \bar{w}_k.$$

• We can create a Kalman filter for this system, provided that the cross-correlation between the new process noise \bar{w}_k and the sensor

noise v_k is zero. We enforce this:

$$\mathbb{E}[\bar{w}_k v_k^T] = \mathbb{E}\left[[w_k - T v_k]v_k^T\right] = \Sigma_{\tilde{w}\tilde{v}} - T\Sigma_{\tilde{v}} = 0.$$

- This gives us that the previously unspecified matrix $T = \Sigma_{\widetilde{w}\widetilde{v}} \Sigma_{\widetilde{v}}^{-1}$.
- Using the above, the covariance of the new process noise may be found to be

$$\begin{split} \Sigma_{\widetilde{w}} &= \mathbb{E}[\bar{w}_k \bar{w}_k^T] \\ &= \mathbb{E}\left[\left[w_k - \Sigma_{\widetilde{w}\widetilde{v}} \Sigma_{\widetilde{v}}^{-1} v_k \right] \left[w_k - \Sigma_{\widetilde{w}\widetilde{v}} \Sigma_{\widetilde{v}}^{-1} v_k \right]^T \right] \\ &= \Sigma_{\widetilde{w}} - \Sigma_{\widetilde{w}\widetilde{v}} \Sigma_{\widetilde{v}}^{-1} \Sigma_{\widetilde{w}\widetilde{v}}^T. \end{split}$$

A new Kalman filter may be generated using these definitions:

$$\bar{A}_{k} = A_{k} - \Sigma_{\tilde{w}\tilde{v}}\Sigma_{\tilde{v}}^{-1}C_{k}$$
$$\Sigma_{\tilde{w}} = \Sigma_{\tilde{w}} - \Sigma_{\tilde{w}\tilde{v}}\Sigma_{\tilde{v}}^{-1}\Sigma_{\tilde{w}\tilde{v}}^{T}$$

and

$$\begin{aligned} x_{k+1} &= (A_k - \Sigma_{\widetilde{w}\widetilde{v}} \Sigma_{\widetilde{v}}^{-1} C_k) x_k + (B_k - \Sigma_{\widetilde{w}\widetilde{v}} \Sigma_{\widetilde{v}}^{-1} D_k) u_k + \Sigma_{\widetilde{w}\widetilde{v}} \Sigma_{\widetilde{v}}^{-1} z_k + \bar{w}_k \\ z_k &= C_k x_k + D_k u_k + v_k. \end{aligned}$$

Cross-correlated process and measurement noises: Shifted

- A close relation to the above is when the process noise and sensor noise have correlation one timestep apart.
- That is, we assume that $\mathbb{E}[w_k w_j^T] = \Sigma_{\widetilde{w}} \delta_{kj}$, $\mathbb{E}[v_k v_j^T] = \Sigma_{\widetilde{v}} \delta_{kj}$, and $\mathbb{E}[w_k v_j^T] = \Sigma_{\widetilde{w}\widetilde{v}} \delta_{k,j-1}$. The cross-correlation is nonzero only between w_{k-1} and v_k .
- We can re-derive the KF equations using this assumption. We will find that the differences show up in the state-error covariance terms.

The state prediction error is

$$\tilde{x}_k^- = x_k - \hat{x}_k^- = A_k \tilde{x}_{k-1}^+ + w_{k-1}.$$

With the assumptions of this section, the covariance between the state prediction error and the measurement noise is

$$\mathbb{E}[\tilde{x}_k^- v_k^T] = \mathbb{E}\left[[A_k \tilde{x}_{k-1}^+ + w_{k-1}]v_k^T\right] = \Sigma_{\tilde{w}\tilde{v}}.$$

The covariance between the state and the measurement becomes

$$\mathbb{E}\left[\tilde{x}_{k}^{-}\tilde{z}_{k}^{T} \mid \mathbb{Z}_{k-1}\right] = \mathbb{E}\left[\tilde{x}_{k}^{-}\left(C_{k}\tilde{x}_{k}^{-}+v_{k}\right)^{T} \mid \mathbb{Z}_{k-1}\right]$$
$$= \Sigma_{\tilde{x},k}^{-}C_{k}^{T}+\Sigma_{\tilde{w}\tilde{v}}.$$

The measurement prediction covariance becomes

$$\begin{split} \Sigma_{\tilde{z},k} &= \mathbb{E}\left[\tilde{z}_k \tilde{z}_k^T\right] \\ &= \mathbb{E}\left[[C_k \tilde{x}_k^- + v_k][C_k \tilde{x}_k^- + v_k]^T\right] \\ &= C_k \Sigma_{\tilde{x},k}^- C_k^T + \Sigma_{\tilde{v}} + C_k \Sigma_{\tilde{w}\tilde{v}} + \Sigma_{\tilde{w}\tilde{v}}^T C_k^T. \end{split}$$

The modified KF gain then becomes,

$$L_{k} = \left[\Sigma_{\tilde{x},k}^{-}C_{k}^{T} + \Sigma_{\tilde{w}\tilde{v}}\right] \left(C_{k}\Sigma_{\tilde{x},k}^{-}C_{k}^{T} + \Sigma_{\tilde{v}} + C_{k}\Sigma_{\tilde{w}\tilde{v}} + \Sigma_{\tilde{w}\tilde{v}}^{T}C_{k}^{T}\right)^{-1}.$$

- Except for the modified filter gain, all of the KF equations are the same as in the standard case.
- Note that since w_{k-1} is the process noise corresponding to the interval [t_{k-1}, t_k] and v_j is the measurement noise at t_j, it can be seen that the first case considered process noise correlated with measurement noise at the beginning of the above interval, and the second case considered process noise correlated with the end of the interval.

Auto-correlated process noise

- Another common situation that contradicts the KF assumptions is that the process noise is correlated in time.
- That is, with the standard assumption that $x_{k+1} = A_k x_k + B_k u_k + w_k$, we do not have zero-mean white-noise w_k .
- Instead, we have $w_k = A_w w_{k-1} + \bar{w}_{k-1}$, where \bar{w}_{k-1} is a zero-mean white-noise process.
- We handle this situation by estimating both the true system state x_k and also the noise state w_k. We have

$$\begin{bmatrix} x_k \\ w_k \end{bmatrix} = \begin{bmatrix} A_{k-1} & I \\ 0 & A_w \end{bmatrix} \begin{bmatrix} x_{k-1} \\ w_{k-1} \end{bmatrix} + \begin{bmatrix} B_{k-1} \\ 0 \end{bmatrix} u_{k-1} + \begin{bmatrix} 0 \\ \bar{w}_{k-1} \end{bmatrix}$$
$$x_k^* = A_{k-1}^* x_{k-1}^* + B_{k-1}^* u_{k-1} + w_{k-1}^*,$$

where the overall process noise covariance is

$$\Sigma_{\tilde{w}^*} = \mathbb{E}[w_{k-1}^*(w_{k-1}^*)^T] = \begin{bmatrix} 0 & 0 \\ 0 & \mathbb{E}[\bar{w}_{k-1}(\bar{w}_{k-1})^T] \end{bmatrix}$$

and the output equation is

$$z_k = \begin{bmatrix} C_k & 0 \end{bmatrix} \begin{bmatrix} x_k \\ w_k \end{bmatrix} + D_k u_k + v_k$$
$$= C_k^* x_k^* + D_k u_k + v_k.$$

 A standard Kalman filter may now be designed using the definitions of x^{*}_k, A^{*}_k, B^{*}_k, C^{*}_k, D_k, Σ_{w̃*}, and Σ_ṽ.

Auto-correlated sensor noise

• Similarly, we might encounter situations with auto-correlated sensor noise: $v_k = A_v v_{k-1} + \bar{v}_{k-1}$, where \bar{v}_k is white.

• We take a similar approach. The augmented system is

$$\begin{bmatrix} x_k \\ v_k \end{bmatrix} = \begin{bmatrix} A_{k-1} & 0 \\ 0 & A_v \end{bmatrix} \begin{bmatrix} x_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} B_{k-1} \\ 0 \end{bmatrix} u_{k-1} + \begin{bmatrix} w_{k-1} \\ \bar{v}_{k-1} \end{bmatrix}$$
$$x_k^* = A_{k-1}^* x_{k-1}^* + B_{k-1}^* u_{k-1} + w_{k-1}^*$$

with output equation

$$z_{k} = \begin{bmatrix} C_{k} & I \end{bmatrix} \begin{bmatrix} x_{k} \\ v_{k} \end{bmatrix} + D_{k}u_{k} + 0$$
$$= C_{k}^{*}x_{k}^{*} + D_{k}u_{k} + 0$$

The covariance of the combined process noise is

$$\Sigma_{\widetilde{w}^*} = \mathbb{E}\left[\left(\begin{array}{cc} w_k \\ \overline{v}_k \end{array}\right) \left(\begin{array}{cc} w_k & \overline{v}_k \end{array}\right)\right] = \left[\begin{array}{cc} \Sigma_{\widetilde{w}} & 0 \\ 0 & \Sigma_{\widetilde{v}} \end{array}\right].$$

A Kalman filter may be designed using these new definitions of of x^{*}_k, A^{*}_k, B^{*}_k, C^{*}_k, D_k, Σ_{w̃*}, with Σ_ṽ = 0 (the placeholder for measurement noise is zero in the above formulations).

Measurement differencing

- Zero-covariance measurement noise can cause numerical issues.
- A sneaky way to fix this is to introduce an artificial measurement that is computed as a scaled difference between two actual measurements: *z
 k* = *z*{k+1} − *A*_v*z*_k.
- KF equations can then be developed using this new "measurement."
- The details are really messy and not conducive to a lecture presentation. I refer you to Bar-Shalom!
- Care must be taken to deal with the "future" measurement z_{k+1} in the update equations, but it works out to a causal solution in the end.

5.6: Kalman-filter prediction and smoothing

 Prediction is the estimation of the system state at a time *m* beyond the data interval. That is,

$$\hat{x}_{m|k}^{-} = \mathbb{E}[x_m \mid \mathbb{Z}_k],$$

where m > k.

- There are three different prediction scenarios:
 - <u>Fixed-point prediction</u>: Find $\hat{x}_{m|k}^{-}$ where *m* is fixed, but *k* is changing as more data becomes available;
 - Fixed-lead prediction: Find $\hat{x}_{k+L|k}^{-}$ where L is a fixed lead time;
 - <u>Fixed-interval prediction</u>: Find $\hat{x}_{m|k}^-$ where k is fixed, but m can take on multiple future values.
- The desired predictions can be extrapolated from the standard Kalman filter state and estimates.
- The basic approach is to use the relationship (cf. Homework 1)

$$x_{m} = \left(\prod_{j=0}^{m-k-1} A_{m-1-j}\right) x_{k} + \sum_{i=k}^{m-1} \left(\prod_{j=0}^{m-i-2} A_{m-1-j}\right) B_{i} u_{i}$$
$$+ \sum_{i=k}^{m-1} \left(\prod_{j=0}^{m-i-2} A_{m-1-j}\right) w_{i},$$

in the relationship

$$\hat{x}_{m|k}^{-} = \mathbb{E}[x_m \mid \mathbb{Z}_k],$$

with the additional knowledge that $\hat{x}_k^+ = \mathbb{E}[x_k \mid \mathbb{Z}_k]$ from a standard Kalman filter.

That is,

$$\begin{aligned} \hat{x}_{m|k}^{-} &= \mathbb{E}[x_{m} \mid \mathbb{Z}_{k}] \\ &= \mathbb{E}\left[\left(\prod_{j=0}^{m-k-1} A_{m-1-i} \right) x_{k} \mid \mathbb{Z}_{k} \right] + \mathbb{E}\left[\sum_{i=k}^{m-1} \left(\prod_{j=0}^{m-i-2} A_{m-1-i} \right) B_{i}u_{i} \mid \mathbb{Z}_{k} \right] \\ &+ \mathbb{E}\left[\sum_{i=k}^{m-1} \left(\prod_{j=0}^{m-i-2} A_{m-1-j} \right) w_{i} \mid \mathbb{Z}_{k} \right] \\ &= \left(\prod_{j=0}^{m-k-1} A_{m-1-i} \right) \hat{x}_{k}^{+} + \sum_{i=k}^{m-1} \left(\prod_{j=0}^{m-i-2} A_{m-1-i} \right) B_{i}\mathbb{E}[u_{i} \mid \mathbb{Z}_{k}]. \end{aligned}$$

- Note that we often assume that $\mathbb{E}[u_k] = 0$.
- If, furthermore, the system is time invariant,

$$\hat{x}_{m|k}^- = A^{m-k}\hat{x}_k^+.$$

The covariance of the prediction is

$$\Sigma_{\tilde{x},m|k}^{-} = \mathbb{E}[(x_m - \hat{x}_{m|k}^{-})(x_m - \hat{x}_{m|k}^{-})^T \mid \mathbb{Z}_k]$$

= $\left(\prod_{j=0}^{m-k-1} A_{m-1-i}\right) \Sigma_{\tilde{x},k}^{+} \left(\prod_{j=0}^{m-k-1} A_{m-1-i}\right)^T$
+ $\sum_{i=k}^{m-1} \left(\prod_{j=0}^{m-i-2} A_{m-1-j}\right) \Sigma_{\tilde{w}} \left(\prod_{j=0}^{m-i-2} A_{m-1-j}^T\right).$

If the system is time invariant, this reduces to

$$\Sigma_{\tilde{x},m|k}^{-} = A^{m-k} \Sigma_{\tilde{x},k}^{+} \left(A^{m-k}\right)^{T} + \sum_{j=1}^{m-k} A^{j} \Sigma_{\widetilde{w}} \left(A^{j}\right)^{T}.$$

Smoothing

 Smoothing is the estimation of the system state at a time *m* amid the data interval. That is,

$$\hat{x}_{m|N}^+ = \mathbb{E}[x_m \mid \mathbb{Z}_N],$$

where m < N.

- There are three different smoothing scenarios:
 - Fixed-point smoothing: Find x⁺_{m|k} where m is fixed, but k is changing as more data becomes available;
 - Fixed-lag smoothing: Find $\hat{x}_{k-L|k}^{-}$ where L is a fixed lag time;
 - <u>Fixed-interval smoothing</u>: Find $\hat{x}_{m|N}^+$ where k is fixed, but m can take on multiple past values.
- Of these, fixed-interval smoothing is the most relevant, and both texts have detailed derivations.
- The others use a variation of this idea.

Fixed interval smoothing

- The algorithm consists of a forward recursive pass followed by a backward pass.
- The forward pass uses a Kalman filter, and saves the intermediate results \hat{x}_k^- , \hat{x}_k^+ , $\Sigma_{\tilde{x},k}^-$, and $\Sigma_{\tilde{x},k}^+$.
- The backward pass starts at time N of the last measurement, and computes the smoothed state estimate using the results obtained from the forward pass.

Recursive equations (backward sweep)

$$\hat{x}_{m|N}^{+} = \hat{x}_{m}^{+} + \lambda_{m} \left(\hat{x}_{m+1|N}^{+} - \hat{x}_{m+1}^{-} \right)$$
$$\lambda_{m} = \Sigma_{\tilde{x},m}^{+} A_{m}^{T} \left(\Sigma_{\tilde{x},m+1}^{-} \right)^{-1}$$

where m = N - 1, N - 2, ..., 0. Note, $\hat{x}_{N|N}^+ = \hat{x}_N^+$ to start backward pass.

The error covariance matrix for the smoothed estimate is

$$\Sigma_{\tilde{x},m|N}^{+} = \Sigma_{\tilde{x},m}^{+} + \lambda_m \left[\Sigma_{\tilde{x},m+1|N}^{+} - \Sigma_{\tilde{x},m+1}^{-} \right] \lambda_m^T,$$

but it is not needed to be able to perform the backward pass.

Note, the term in the square brackets is negative semi-definite, so the covariance of the smoothed estimate is "smaller" than for the filtered estimate only.

Fixed point smoothing

• Here, m is fixed, and the final point k keeps increasing.

$$\hat{x}_{m|k}^{+} = \hat{x}_{m|k-1}^{+} + \mu_{k} \left(\hat{x}_{k}^{+} - \hat{x}_{k}^{-} \right)$$
$$\mu_{k} = \prod_{i=m}^{k-1} \lambda_{i},$$

where the product multiplies on the left as *i* increases.

• For k = m + 1,

$$\hat{x}_{m|m+1}^{+} = \hat{x}_{m}^{+} + \mu_{m+1} \left(\hat{x}_{m+1}^{+} - \hat{x}_{m+1}^{-} \right)$$
$$\mu_{m+1} = \lambda_{m} = \Sigma_{\tilde{x},m}^{+} A_{m}^{T} \left(\Sigma_{\tilde{x},m+1}^{-} \right)^{-1}$$

• For k = m + 2,

$$\hat{x}_{m|m+2}^{+} = \hat{x}_{m|m+1}^{+} + \mu_{m+2} \left(\hat{x}_{m+2}^{+} - \hat{x}_{m+2}^{-} \right)$$
$$\mu_{m+2} = \sum_{\tilde{x},m+1}^{+} A_{m+1}^{T} \left(\sum_{\tilde{x},m+2}^{-} \right)^{-1} \mu_{k+1},$$

and so forth.

Fixed lag smoothing

- Here, we seek to estimate the state vector at a fixed time interval lagging the time of the current measurement.
- This type of smoothing trades off estimation latency for more accuracy.
- The fixed interval smoothing algorithm could be used to perform fixed-lag smoothing when the number of backward steps equals the time lag
- This is fine as long as the number of backward steps is small.
- Fixed-lag smoothing algorithm has a startup problem: Cannot run until enough data is available.

5.7: Reduced-order Kalman filter

- Why estimate the entire state vector when you are measuring one or more state elements directly?
- Consider partitioning system state into (may require transformation)

 $x_{k:a}$: measured state

 $x_{k:b}$: to be estimated.

■ So,

 $z_k = Cx_k + Du_k + v_k = x_{k:a} + Du_k + v_k$

$$\hat{x}_{k:a} = z_k - Du_k = x_{k:a} + v_k.$$

- We assume that the measurement is noise-free (otherwise, we would need to estimate x_{k:a} also). So, x̂_{k:a} = z_k Du_k = x_{k:a}.
- In order to design an estimator for x_{k:b}, we need to create a suitable state-space model of the dynamics of x_{k:b}.
- We begin by writing the equations for the partitioned system

$$\begin{bmatrix} x_{k+1:a} \\ \hline x_{k+1:b} \end{bmatrix} = \begin{bmatrix} A_{aa} & A_{ab} \\ \hline A_{ba} & A_{bb} \end{bmatrix} \begin{bmatrix} x_{k:a} \\ \hline x_{k:b} \end{bmatrix} + \begin{bmatrix} B_a \\ B_b \end{bmatrix} u_k + \begin{bmatrix} w_{k:a} \\ \hline w_{k:b} \end{bmatrix}$$
$$z_k = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} x_{k:a} \\ \hline x_{k:b} \end{bmatrix} + Du_k.$$

• We wish to write the $x_{k:b}$ dynamics in the form:

$$x_{k+1:b} = A_{xb}x_{k:b} + B_{xb}m_{k,1} + \bar{w}_k$$

$$m_{k,2} = C_{xb} x_{k:b} + D_{xb} m_{k,1} + \bar{v}_k,$$

where $m_{k,1}$ and $m_{k,2}$ are some measurable inputs, and will be combinations of z_k and u_k .

- Once we have this state-space form, we can create a Kalman-filter state estimator for x_{k:b}.
- We start the derivation by finding an output equation for x_{k:b}. Consider the dynamics of the measured state:

$$x_{k+1:a} = A_{aa}x_{k:a} + A_{ab}x_{k:b} + B_{a}u_{k} + w_{k:a}$$
$$z_{k+1} = A_{aa}z_{k} + A_{ab}x_{k:b} + B_{a}u_{k} + w_{k:a}.$$

Let

$$m_{k,2}=z_{k+1}-A_{aa}z_k-B_au_k.$$

Then

$$m_{k,2} = A_{ab} x_{k:b} + w_{k:a},$$

where $m_{k,2}$ is known/measurable and thus " C_{xb} " is equal to A_{ab} .

- This is our reduced-order estimator output relation.
- We now look for a state equation for x_{k:b}. Consider the dynamics of the estimated state:

$$x_{k+1:b} = A_{ba}x_{k:a} + A_{bb}x_{k:b} + B_{b}u_{k} + w_{k:b}$$
$$A_{bb}x_{k:b} + A_{ba}z_{k} + B_{b}u_{k} + w_{k:b}.$$

Let

$$B_{xb}m_{k,1} = A_{ba}z_k + B_bu_k$$

so that the reduced-order recurrence relation is

$$x_{k+1:b} = A_{bb} x_{k:b} + B_{xb} m_{k,1} + w_{k:b}.$$

This might be accomplished via

$$B_{xb}m_{k,1} = \left[A_{ba} B_b\right] \left[\begin{matrix} z_k \\ u_k \end{matrix}\right]$$

although the details of how this is done do not matter in the end.

So, for the purpose of designing our estimator, the state-space equations are:

$$x_{k+1:b} = A_{bb} x_{k:b} + B_{xb} m_{k,1} + \bar{w}_k$$

$$m_{k,2} = A_{ab} x_{k:b} + \bar{v}_k,$$

where $\bar{w}_k = w_{k:b}$ and $\bar{v}_k = w_{k:a}$.

- Note that the measurement is non-causal, so the filter output will lag the true output by one sample.
- Another (causal) approach that does not assume noise-free measurements is presented in: D. Simon, "Reduced Order Kalman Filtering without Model Reduction," *Control and Intelligent Systems*, vol. 35, no. 2, pp. 169–174, April 2007.
- This algorithm can end up more complicated than full Kalman filter unless many states are being removed from estimation requirements.

5.8: Measurement validation gating

- Sometimes the systems for which we would like a state estimate have sensors with intermittent faults.
- We would like to detect faulty measurements and discard them (the time update steps of the KF are still implemented, but the measurement update steps are skipped).
- The Kalman filter provides an elegant theoretical means to accomplish this goal. Note:
 - The measurement covariance matrix is $\Sigma_{\tilde{z},k} = C_k \Sigma_{\tilde{x},k}^- C_k^T + \Sigma_{\tilde{v}};$
 - The measurement prediction itself is $\hat{z}_k = C_k \hat{x}_k^- + D_k u_k$;
 - The innovation is $\tilde{z}_k = z_k \hat{z}_k$.
- A measurement validation gate can be set up around the measurement using <u>normalized estimation error squared</u> (NEES)

$$e_k^2 = \tilde{z}_k^T \Sigma_{\tilde{z},k}^{-1} \tilde{z}_k.$$

- NEES e_k^2 varies as a Chi-squared distribution with *m* degrees of freedom, where *m* is the dimension of z_k .
- If e_k^2 is outside the bounding values from the Chi-squared distribution for a desired confidence level, the measurement is discarded.
- Note: If a number of measurements are discarded in a short time interval, it may be that the sensor has truly failed, or that the state estimate and its covariance has gotten "lost."
- It is sometimes helpful to "bump up" the covariance Σ[±]_{x̃,k}, which simulates additional process noise, to help the Kalman filter to reacquire.

Chi-squared test

 A chi-square random variable is defined as a sum of squares of independent unit variance zero mean normal random variables.

$$Y = \sum_{i=1}^{n} \left(\frac{X_i - \mathbb{E}[X_i]}{\sigma} \right)^2.$$

- *Y* is *chi-square* with *n* degrees of freedom.
- Since it is a sum of squares, it is never negative and is not symmetrical about its mean value.
- The pdf of Y with n degrees of freedom is

$$f_Y(y) = \frac{1}{2^{n/2}\Gamma(n/2)} y^{(n/2-1)} e^{-n/2}.$$

• For confidence interval estimation we need to find two critical values.



Critical values for 95% confidence and χ^2 with 24 degrees of freedom

If we want 1 − α confidence that the measurement is valid, we want to make sure that *Y* is between χ²_L and χ²_U where χ²_L is calculated using α/2 and χ²_U is calculated using 1 − α/2.

- The χ^2 pdf, cdf, and inverse cdf are available in most analysis software packages (*e.g.*, MATLAB, Mathematica, and even the spreadsheet program Excel).
- For hand calculations a χ^2 -table is available on page ??.

TRICKS WITH MATLAB: MATLAB may also be used to find the χ^2 values if a table is not convenient. This requires the MATLAB *statistics toolbox:*

>> help chi2inv

```
CHI2INV Inverse of the chi-square cumulative distribution function (cdf).

X = CHI2INV(P,V) returns the inverse of the chi-square cdf with V

degrees of freedom at the values in P. The chi-square cdf with V

degrees of freedom, is the gamma cdf with parameters V/2 and 2.

The size of X is the common size of P and V. A scalar input

functions as a constant matrix of the same size as the other input.
```

```
>> % want to compute values for alpha = 0.05
>> chi2inv(1-.025,24) % Tail probability of alpha/2=0.025, n = 24. Upper
critical value
```

ans = 39.3641

>> chi2inv(.025,24) % Lower critical value

ans = 12.4012

Appendix: Critical Values of χ^2

For some deg. of freedom, each entry represents the critical value of χ² for a specified upper tail area α.

|)5 | 61 | ⁶⁷ | 8
3
3
3 | 60 | 20 | 48 | 78
A | ۲ | 68 | 0
00
00 | 57 | 00 | 19 | 19 | 01 | 67 | 0 1 | T O | то
56 | 8 2 6
8 2 | 56
82
97 | то
56
97
01 | 56
97
96
11 | 8 0 1 2 2 0 2 2 0 2 1 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 2 12 18 20 2 18 19 19 18 18 18 18 18 18 18 18 18 18 18 18 18 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 4 9 2 9 9 9 9 8 9 9 9
5 0 8 9 1 7 2 8 6 9
0
 | 9 4 9 0 8 9 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | а а а а а а а а а а а а а а а а а а а | 7 m 6 4 6 0 8 8 0 1 7 1 0 0 0 0 7 0 0 0 7 0 0 0 1 7 0 0 0 1 7 0 0 0 1 7 0 0 0 1 7 0 0 0 0
 | 0 7 8 6 7 7 8 6 7 7 8 6 7 7 8 6 7 9 8 6 7 7 9 8 6 7 7 9 8 6 7 9 7 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 | 0
0
0 7 9 8 0 0 1 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 | 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 | 0 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 0 χ ² _{U(α,df)}
0 χ ² _{U(α,df)} | 0
2
2
2
2
2
2
2
2
2
2
2
2
2
 | 0 χ ² _{U(a,df)}
0 χ ² _{U(a,df)}
0 χ ² _{U(a,df)} | 0 χ ² _{U(α,df)}
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 0 χ ² _{U(α,df)}
0 χ ² _{U(α,df)}
 |
|-------------------|-------|---------------|------------------|--------|--------|--------|---------|----------|--------|---------------|--------|--------|--------|--------|--------|--------|--------|--------|----------------|------------------|----------------------------|--|---|---|---|--|--
--|--
--
---|--
--
--	---
--	---

0.0(7.8
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
200
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2000
2 | 200
200
200
200
200
200
200
200 | 0 0 <td>30 <td< td=""><td>30 <</td><td>0 0<td>0 0<td></td><td></td><td>0 0<td>0 0<td>0 0<td>0 0
 0 0</td><td>0 0</td></td></td></td></td></td></td<></td> | 30 30 <td< td=""><td>30 <</td><td>0 0<td>0 0<td></td><td></td><td>0 0<td>0 0<td>0 0<td>0 0</td><td>0 0</td></td></td></td></td></td></td<> | 30 30
 30 < | 0 0 <td>0 0<td></td><td></td><td>0 0<td>0 0<td>0 0<td>0 0</td><td>0 0</td></td></td></td></td> | 0 0 <td></td> <td></td> <td>0 0<td>0 0<td>0 0<td>0 0
 0 0</td><td>0 0</td></td></td></td> | | | 0 0 <td>0 0<td>0 0<td>0 0
0 0 0 0 0 0 0 0 0 0 0</td><td>0 0</td></td></td> | 0 0 <td>0 0<td>0 0</td><td>0 0</td></td> | 0 0 <td>0 0</td> <td>0 0</td> | 0
 | 0 |
| 0.01 | 6.635 | 9.210 | 11.345 | 13.277 | 15.086 | 16.812 | 18.475 | 20.090 | 21.666 | 23.209 | 24.725 | 26.217 | 27.688 | 29.141 | 30.578 | 32.000 | 33.409 | | 34.805 | 34.805
36.191 | 34.805
36.191
37.566 | 34.805
36.191
37.566
38.932 | 34.805
36.191
37.5666
38.9328
40.289 | 34.805
36.191
37.5666
38.932
40.289
41.638 | 34.805
36.1911
37.566
38.932
38.932
40.289
41.638
42.980 | 34.805
36.191
37.566
38.932
38.932
40.289
41.638
42.980
42.980 | 34.805
36.1911
37.566
38.932
40.289
41.638
42.980
44.314
45.642 | 34.805
36.1911
36.1913
37.566
38.932
40.289
41.638
41.638
42.980
42.980
45.642
45.642
46.963
 | 34.805
36.191
36.191
37.566
40.289
41.638
42.987
44.314
45.638
45.642
46.963
48.278 | 34.805
36.191
37.566
40.289
41.638
42.980
42.980
42.980
45.642
45.642
46.963
48.278
49.588 | 34.805
36.1919
37.566
40.289
41.638
42.980
42.980
45.642
45.642
45.642
45.642
45.642
45.642
45.642
49.588
49.588
49.5892
50.892
 | 34.805
36.191
37.566
40.289
41.638
42.982
44.314
45.642
45.642
46.963
46.963
48.278
48.278
49.583
49.5938
50.892
51911 | 34.805
36.191
36.191
37.566
40.289
41.638
42.980
42.980
45.642
46.963
48.278
48.278
48.278
48.278
50.892
53.486
53.486
 | 34.805
36.191
37.566
40.289
41.638
42.980
42.980
44.314
45.642
48.278
49.588
49.588
49.588
53.486
53.486 | 34.805
36.191
37.566
40.289
41.638
42.980
42.980
45.642
48.278
48.278
49.588
49.588
49.588
50.892
53.491
54.776
56.061
 | 34.805
36.191
37.566
41.638
42.932
42.986
44.314
45.642
45.642
46.963
45.642
46.963
45.642
46.963
45.642
46.963
50.892
50.892
53.496
54.776
55.061
55.061
56.061
57.342
56.061
57.342
56.061
57.342
56.061
57.342
56.061
57.342
56.061
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.342
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57.3425
57. | 34.805
36.191
36.191
37.566
40.289
40.289
42.989
45.642
45.642
45.642
46.963
50.892
50.892
53.486
53.486
53.486
53.486
53.3486
53.3486
54.776
55.061
55.061
55.061
56.061 | 34.805
36.191
36.191
37.566
40.289
41.638
42.980
42.980
45.642
46.963
46.963
50.892
50.892
53.486
54.776
53.191
55.191
56.963
56.061
57.342
58.619
58.619
58.619
58.619
59.893
 | 34.805
36.191
36.191
37.566
40.289
41.638
42.980
42.980
42.980
42.980
42.980
42.980
42.980
42.980
42.980
42.980
42.980
53.486
53.486
53.486
55.061
58.051
58.051
58.051
58.051
58.051
58.051
58.051
58.051
58.051
58.051
58.051
58.051
58.051
58.051
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.050
58.0500
58.0500
58.0500
58.0500
58.0500
58.05000
58.050000000000 | 34.805
36.1905
37.5669
41.63893
42.980
42.980
42.980
45.642
46.963
46.963
56.248
46.963
56.25
56.25
56.25
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56.35
56 |
34.805
36.905
37.5566
41.638932.932
42.986
42.986
42.986
45.642
45.642
46.963
56.061
56.061
56.061
57.342
56.061
56.061
56.061
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.62
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061
56.061000000000000000000 |
| 0.025 | 5.024 | 7.378 | 9.348 | 11.143 | 12.833 | 14.449 | 16.013 | 17.535 | 19.023 | 20.483 | 21.920 | 23.337 | 24.736 | 26.119 | 27.488 | 28.845 | 30.191 | | 31.526 | 31.526
32.852 | 31.526
32.852
34.170 | 31.526
32.852
34.170
35.479 | 31.526
32.852
34.170
35.479
35.781 | 31.526
32.852
34.170
35.479
36.781
38.076 | 31.526
32.852
34.170
35.479
36.781
38.076
39.364 | 31.526
32.852
34.170
35.479
36.781
38.076
39.364
40.646 | 31.526
32.852
34.170
35.479
35.479
35.781
38.076
39.364
40.646
41.923 | 31.526
32.852
34.170
35.479
36.781
38.076
39.364
40.646
41.923
41.95
 | 31.526
32.852
34.170
35.479
36.781
38.076
39.364
41.923
43.195
44.461 | 31.526
32.852
34.170
35.479
36.781
38.076
39.364
41.923
43.195
44.461
44.461 | 31.526
32.852
34.170
35.479
35.479
35.781
39.364
40.646
40.646
41.923
43.195
44.461
45.722
45.722
 | 31.526
32.852
34.170
35.479
36.781
38.7781
38.076
39.364
41.923
41.923
43.195
44.461
45.722
46.979
46.979 | 31.526
32.852
34.170
35.479
36.781
38.076
39.364
41.923
43.195
44.461
44.461
44.461
45.722
46.979
48.232
48.232
 | 31.526
32.852
34.170
35.479
35.479
36.781
38.076
39.364
40.646
41.923
43.195
44.461
44.461
45.722
46.979
48.232
49.480
50.725
50.725 | 31.526
32.852
34.170
35.479
35.479
36.781
38.076
39.364
40.646
41.923
44.461
44.461
45.722
45.722
45.79
48.232
49.480
50.725
51.966
 | 31. 526
32. 852
34. 170
35. 479
35. 479
35. 479
39. 364
40. 646
40. 646
41. 923
43. 195
44. 461
45. 722
49. 480
48. 232
49. 480
50. 725
51. 966
51. 966 | 31. 526
32. 852
34. 170
35. 479
35. 479
35. 479
35. 479
39. 364
40. 646
41. 923
41. 923
41. 923
41. 923
42. 195
44. 461
45. 725
49. 480
50. 725
51. 966
51. 966
51. 966 | 31. 526
32. 852
34. 170
35. 479
36. 781
38. 076
39. 364
41. 923
43. 195
44. 461
44. 461
44. 461
45. 722
49. 480
48. 232
49. 480
50. 725
51. 966
53. 203
54. 437
55. 668
 | 31. 526
32. 852
34. 170
35. 479
35. 479
36. 781
38. 076
39. 364
41. 923
44. 461
44. 461
44. 461
44. 461
44. 461
45. 722
49. 480
49. 480
50. 725
51. 966
53. 203
54. 437
55. 668 | 31. 526
32. 852
34. 170
35. 479
35. 479
36. 781
38. 076
39. 364
40. 646
41. 923
44. 461
45. 722
44. 461
45. 722
44. 461
45. 722
51. 966
51. 966
51. 966
51. 966
51. 966
51. 966
55. 668
55. 868
58. 120 | 31. 526
32. 852
34. 170
35. 479
35. 479
35. 479
39. 364
40. 646
41. 923
43. 195
44. 461
45. 722
49. 480
50. 725
51. 966
51. 966
51. 966
51. 966
51. 966
51. 966
53. 203
56. 896
58. 120
59. 342
 |
| 0.05 | 3.841 | 5.991 | 7.815 | 9.488 | 11.070 | 12.592 | 14.067 | 15.507 | 16.919 | 18.307 | 19.675 | 21.026 | 22.362 | 23.685 | 24.996 | 26.296 | 27.587 | | 28.869 | 28.869
30.144 | 28.869
30.144
31.410 | 28.869
30.144
31.410
32.671 | 28.869
30.144
31.410
32.671
33.924 | 28.869
30.144
31.410
32.671
33.924
35.172 | 28.869
30.144
31.410
32.671
33.924
35.172
36.415 | 28.869
30.144
31.410
32.671
33.924
35.172
35.415
36.415
37.652 | 28.869
30.144
31.410
32.671
33.924
35.172
36.415
36.415
37.652
38.885 | 28.869
30.144
31.410
32.671
33.924
33.172
35.172
36.415
37.652
38.885
40.113
 | 28.869
30.144
31.410
32.671
33.924
35.172
35.415
35.415
35.415
38.855
38.855
40.113
41.337 | 28.869
30.144
31.410
32.671
33.924
35.172
35.415
35.415
36.415
38.885
38.885
40.113
41.337 | 28.869
30.144
31.410
32.671
35.172
35.172
35.415
36.415
37.652
37.652
38.885
40.113
41.337
42.557
43.773
 | 28.869
30.144
31.410
32.671
33.924
35.172
35.415
36.415
37.652
385
40.113
42.557
42.557
43.773 | 28.869
30.144
31.410
32.671
33.924
35.172
35.415
35.415
35.415
35.885
40.113
41.337
42.557
42.557
42.557
44.985
 | 28.869
30.144
31.410
32.671
35.172
35.415
35.415
36.415
36.415
37.652
38.885
40.113
41.337
42.557
42.557
42.557
44.985
44.985
44.985 | 28.869
30.144
31.410
32.671
35.172
35.172
35.415
35.415
36.415
37.652
38.885
40.113
41.337
42.557
41.337
42.557
41.337
42.557
41.337
42.602
48.602
 | 28.869
30.144
31.410
32.671
35.172
35.172
35.172
35.172
35.172
35.172
415
415
41.337
42.557
42.557
42.557
42.557
42.557
42.985
44.985
42.602
47.400
48.602
48.602 | 28.869
30.144
31.410
32.671
33.924
35.172
35.172
35.172
35.172
35.172
41.337
41.337
42.337
42.935
44.985
44.985
44.985
42.733
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.998
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.985
42.9856
42.9856
42.9856
42.9856
42.98566
42.9856666666666666666666666666 | 28.869
30.144
31.410
32.671
33.924
35.172
35.415
35.415
36.415
36.415
37.652
38.885
40.113
41.337
42.557
42.557
42.557
42.948
42.955
42.9557
42.9557
42.998
502998
 | 28.869
31.410
32.671
33.924
35.172
35.172
36.415
36.415
36.415
37.652
38.885
40.113
41.337
42.557
41.337
42.557
42.985
44.985
45.194
46.194
47.400
48.602
48.602
48.602
50.998
52.192 | 28.869
30.144
31.410
32.671
35.172
35.172
35.172
36.415
36.415
37.652
38.885
41.337
42.557
41.337
42.557
41.337
42.557
42.602
44.985
44.985
45.194
47.400
48.602
55.3384
55.3384 | 28.869
30.144
31.410
32.671
35.172
35.172
35.172
36.415
37.652
37.652
38.885
41.337
41.337
42.557
42.193
44.985
44.985
44.985
42.192
48.602
48.602
49.802
53.384
55.758
 |
| 0.10 | 2.706 | 4.605 | 6.251 | 7.779 | 9.236 | 10.645 | 12.017 | 13.362 | 14.684 | 15.987 | 17.275 | 18.549 | 19.812 | 21.064 | 22.307 | 23.542 | 24.769 | | 25.989 | 25.989
27.204 | 25.989
27.204
28.412 | 25.989
27.204
28.412
29.615 | 25.989
27.204
28.412
29.615
30.813 | 25.989
27.204
28.412
29.615
30.813
32.007 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
33.196 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
33.196
33.3563 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
34.382
35.563
36.741
 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
33.196
34.382
35.563
36.741
35.916 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
33.196
33.382
35.563
35.741
37.916
37.916 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
33.196
34.382
35.563
35.763
35.741
37.916
39.087
40.256
 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
33.196
34.382
35.563
35.563
35.741
37.916
39.087
40.256
41.422 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
34.382
35.563
35.741
35.741
35.741
35.741
35.916
39.087
41.422
41.422
42.585
 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
33.196
34.382
35.563
35.741
37.916
39.087
41.422
42.585
43.745 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
34.382
35.563
36.741
37.916
39.087
40.256
41.422
43.745
43.745
 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
34.382
35.563
35.761
35.741
37.916
40.256
41.422
41.422
41.422
42.585
41.422
44.903
46.059 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
34.382
35.563
35.741
35.563
35.741
37.916
41.422
41.422
41.422
41.422
41.422
41.422
41.422
41.422
41.422
41.422
41.422
41.256
41.256
41.256
41.256
41.256
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.585
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.556
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5567
42.5577
42.55677
42.55677777777777777777777777777777777777 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
34.382
35.563
35.741
37.916
39.087
41.422
41.422
41.422
41.422
42.585
44.903
46.059
48.363
 | 25.989
27.204
28.412
29.615
30.813
32.007
33.196
34.382
35.563
36.741
35.563
36.741
41.422
41.422
42.585
41.422
42.585
41.422
42.585
42.256
42.256
44.903
48.363
49.513 | 25.989
27.204
28.412
29.615
30.813
32.007
32.007
33.196
34.382
35.563
36.741
37.916
40.256
41.422
41.422
42.585
42.585
44.903
44.903
46.059
48.363
49.513
49.513
660 | 25.989
27.204
28.412
29.615
30.813
32.007
32.007
33.196
34.382
35.563
35.741
37.916
41.422
41.422
41.422
41.422
42.585
44.903
46.059
44.903
46.059
48.363
49.212
48.363
40.256
713
805
50.660
 |
| il Areas
0.25 | 1.323 | 2.773 | 4.108 | 5.385 | 6.626 | 7.841 | 9.037 | 10.219 | 11.389 | 12.549 | 13.701 | 14.845 | 15.984 | 17.117 | 18.245 | 19.369 | 20.489 | | 21.605 | 21.605
22.718 | 21.605
22.718
23.828 | 21.605
22.718
23.828
24.935 | 21.605
22.718
23.828
24.935
26.039 | 21.605
22.718
23.828
24.935
26.039
27.141 | 21.605
22.718
23.828
24.935
26.039
27.141
28.241 | 21.605
22.718
23.828
24.935
24.935
24.935
24.039
27.141
28.241
28.241
28.241 | 21.605
22.718
23.828
24.935
26.039
26.039
27.141
28.241
28.241
29.339
30.435 | 21.605
22.718
23.828
24.935
26.039
26.039
26.039
28.241
28.241
28.241
29.339
30.435
 | 21.605
22.718
23.828
24.935
26.039
26.039
26.039
26.039
28.241
28.241
28.241
28.241
28.243
30.435
31.528
31.528 | 21.605
22.718
23.828
24.935
24.935
26.039
27.141
28.241
28.241
28.241
28.233
33.435
31.528
31.528
33.711 | 21.605
22.718
23.828
24.935
26.039
26.039
27.141
28.241
28.241
29.339
30.435
30.435
31.528
31.528
33.711
33.711
 | 21.605
22.718
23.828
24.935
26.039
26.039
27.141
28.241
28.241
29.339
30.435
30.435
31.528
31.528
31.528
33.711
34.800
33.887 | 21.605
22.718
23.828
24.935
26.039
26.039
27.141
28.241
28.241
28.241
28.241
33.339
31.528
31.528
31.528
33.711
34.887
35.887
35.973
 | 21.605
22.718
23.828
24.935
26.039
26.039
27.141
28.241
28.241
28.241
23.933
31.528
31.528
31.528
31.528
31.528
33.711
34.800
35.887
38.058 | 21.605
22.718
23.828
24.935
24.935
26.039
27.141
28.241
28.241
28.241
28.233
30.435
30.435
31.528
31.528
33.711
33.711
33.620
33.711
33.711
33.711
33.711
33.711
33.711
33.711
 | 21.605
22.718
23.828
24.935
26.039
26.039
27.141
28.241
28.241
28.241
29.339
30.435
30.435
31.528
31.528
31.528
33.711
34.800
35.887
33.711
34.800
33.711
34.973
38.058
39.141 | 21.605
22.718
23.828
24.935
24.935
26.039
27.141
28.241
29.339
30.435
31.528
31.528
31.528
33.711
34.800
33.711
34.800
33.711
34.800
33.711
34.800
33.711
34.800
33.711
34.800
33.711
34.800
33.711
34.800
33.711
34.800
33.711
34.800
33.711
34.800
33.711
34.800
33.7111
34.800
33.7111
34.800
33.7111
34.800
33.7111
34.800
33.7111
34.800
33.7111
34.800
33.7111
34.800
33.7111
34.800
33.71111
34.800
33.7111
34.800
33.71111
34.800
33.71111
34.800
33.71111
34.800
33.71111
34.800
33.71111
34.800
33.711111
34.800
33.71111
34.900
33.711111
34.800
33.721111
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7212
34.800
33.7222
33.7212
34.8000
33.7222
33.7212
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.7222
33.72223
33.72223
33.72223
33.72223
33.72223
33.72223
33.72223
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.72233
33.722333
33.722333
33.7223333
33.72233333
33.7223333333333 | 21.605
22.718
23.828
24.935
24.935
26.039
26.039
27.141
28.241
28.241
28.241
28.243
31.528
31.528
33.711
34.800
35.973
35.973
36.973
36.973
38.058
39.141
41.304
41.304
 | 21.605
22.718
23.828
24.935
24.935
26.039
26.039
27.141
28.241
28.241
28.241
28.243
31.528
31.528
31.528
33.711
34.80
33.711
34.80
33.711
34.87
35.887
33.711
34.87
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.973
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.87735.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.887
35.877
35.877
35.877
35.8772
35.8772
35.8772
35.8772
35.8772
35.87723
35.87723
35.87723
35.87723
35.87723
35.87723
35.87723
35.87723
35.87723
35.87723
35.87723
35.87723
35.87723
35.87723
35.87723
37.87723
37.87723
37.87723
37.87723
37.87723
37.87723
37.87723
37.87723
37.87723
37.87723
37.87723
37.87723
37.87723
37.87720 | 21.605
22.718
23.828
24.935
26.039
26.039
27.141
28.241
28.241
28.241
28.241
28.241
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.711
33.7111
33.7111
33.7111
33.7111
33.7111
33.7123
33.7111
33.7111
33.7123
33.7111
33.7111
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.71233
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.7123
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.712333
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.71233
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.712333
33.7123333
33.7123333
33.71233333
33.7123333333333333333333333333333333333 | 21.605
22.718
23.828
24.935
26.039
27.141
28.241
28.241
29.339
30.435
31.528
31.528
33.711
34.800
33.620
33.711
34.35
39.141
41.304
42.33
41.304
44.539
45.616
 |
| Upper Ta
0.75 | 0.102 | 0.575 | 1.213 | 1.923 | 2.675 | 3.455 | 4.255 | 5.071 | 5.899 | 6.737 | 7.584 | 8.438 | 9.299 | 10.165 | 11.037 | 11.912 | 12.792 | 10 61 | C/0.CT | 14.562 | 13.0/3
14.562
15.452 | 15.9/3
14.562
15.452
16.344 | 15.0/3
14.562
15.452
16.344
17.240 | 15.0/3
14.562
15.452
16.344
17.240
18.137 | 15.0/5
14.562
15.452
16.344
16.344
17.240
18.137
19.037 | 15.0/5
14.562
15.452
16.344
17.240
18.137
19.037
19.939 | 15.0/0
14.562
15.452
16.344
17.240
18.137
19.037
19.939
20.843 | 15.0/0
14.562
15.452
16.344
17.240
18.137
19.037
19.037
19.939
20.843
21.749
 | 15.0.01
14.562
15.452
16.344
17.240
18.137
19.037
19.939
20.843
21.749
22.657 | 15.0.01
14.562
15.452
16.344
17.240
18.137
19.037
19.939
20.843
21.749
22.657
23.567 | 15.0.01
14.562
15.452
16.344
17.240
18.137
19.037
19.037
19.037
19.037
20.843
20.843
20.843
21.749
22.657
23.567
23.567
23.567
 | 15.0.01
14.562
15.452
16.344
17.240
18.137
19.037
19.037
19.037
19.037
20.843
20.843
21.749
21.749
22.657
22.657
23.567
23.567
25.390 | 15.0.01
14.562
15.452
16.344
17.240
18.137
19.037
19.037
19.037
19.037
19.037
21.749
21.749
21.749
22.657
22.657
22.657
22.657
22.390
26.304
 | 15.0.01
14.562
15.452
16.344
17.240
18.137
19.037
19.037
19.037
19.037
19.037
21.749
21.749
22.657
22.657
22.657
22.458
22.657
22.657
22.657
22.5390
22.5390 | 15.0.01
14.562
15.452
16.344
17.240
18.137
19.037
19.037
19.037
19.037
19.037
19.037
21.749
22.657
22.657
22.657
22.657
22.657
22.5390
22.5390
22.5136
22.136
 | 15.0.01
14.562
15.452
16.344
17.240
19.037
19.037
19.037
19.037
19.037
21.749
21.749
21.749
22.657
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
22.572
23.567
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5725
22.5755
22.57555
22.5755557
22.5755575757575757575757575757575757575 | 15.0.01
14.562
15.452
16.344
17.240
18.137
19.037
19.037
19.037
19.037
19.037
21.749
21.749
22.657
22.657
22.657
22.657
22.657
22.657
22.5390
226.304
227.219
28.136
28.136
2973
2973 | 15.0.01
14.562
15.452
16.344
17.240
18.137
19.037
19.037
19.037
19.037
21.749
21.749
22.657
22.657
22.657
22.657
22.390
22.5390
22.27.219
25.390
22.27.219
22.27.219
22.054
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
267
27.219
27.219
27.219
27.2136
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.219
27.213
27.219
27.219
27.229
27.219
27.219
27.219
27.229
27.219
27.239
27.219
27.239
28.230
28.230
28.230
28.230
28.2300
28.2300
 | 15.0.01
14.562
15.452
16.344
17.240
18.137
19.037
19.037
19.037
19.037
21.749
22.657
22.657
22.478
22.657
22.657
22.657
22.19
25.390
25.304
25.304
25.390
267
27.219
28.136
28.136
2973
30.893 | 15.0.01
14.562
15.452
17.240
18.137
19.037
19.037
19.037
19.037
19.037
21.749
22.657
22.657
22.657
22.657
22.657
22.657
22.136
22.136
22.219
22.219
22.219
22.219
22.219
22.219
22.219
22.219
22.219
22.219
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2136
22.2137
22.2136
22.2236
22.2237
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.2236
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.22375
22.2237575
22.2237575757575757575757757757757757775777 | 15.0.01
14.562
15.452
16.344
17.240
18.137
19.037
19.037
19.037
19.037
19.037
22.657
22.657
23.567
22.657
23.567
22.304
22.304
22.304
22.304
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.567
23.577
23.567
23.577
23.577
23.577
23.567
23.577
23.567
23.577
23.577
23.577
23.577
23.577
23.577
23.577
23.577
23.577
23.577
23.577
23.577
23.577
23.5777
23.5777
23.5777
23.5777
23.5777
23.57777
23.5777777777777777777777777777777777777
 |
| 06.0 | 0.016 | 0.211 | 0.584 | 1.064 | 1.610 | 2.204 | 2.833 | 3.490 | 4.168 | 4.865 | 5.578 | 6.304 | 7.042 | 7.790 | 8.547 | 9.312 | 10.085 | | 10.865 | 10.865
11.651 | 10.865
11.651
12.443 | 10.865
11.651
12.443
13.240 | 10.865
11.651
12.443
13.240
14.041 | 10.865
11.651
12.443
13.240
14.041
14.848 | 10.865
11.651
12.443
13.240
14.041
14.848
15.659 | 10.865
11.651
12.443
13.240
14.041
14.848
15.659
15.659
16.473 | 10.865
11.651
12.443
13.240
14.041
14.848
15.659
16.473
17.292 | 10.865
11.651
12.443
13.240
14.041
14.848
15.659
16.473
17.292
18.114
 | 10.865
11.651
12.443
13.240
14.041
14.848
15.659
15.659
16.473
17.292
18.114
18.114
18.939 | 10.865
11.651
12.443
13.240
14.848
15.659
15.659
15.659
15.473
15.292
18.114
18.114
18.939
1939
1939 | 10.865
11.651
12.443
13.240
14.848
15.659
16.473
16.473
16.473
16.473
16.473
16.473
16.939
18.939
19.768
19.768
 | 10.865
11.651
12.443
13.240
14.041
14.848
15.659
16.473
17.292
18.939
18.939
19.768
20.599
21.434 | 10.865
11.651
12.443
13.240
14.041
14.041
14.848
15.659
15.659
17.292
18.114
18.114
18.114
18.939
19.768
20.599
21.434
22.271
 | 10.865
11.651
12.443
13.240
14.041
14.848
15.659
15.659
15.659
17.292
18.114
18.939
1939
1939
1939
20.599
21.434
22.271
22.271 | 10.865
11.651
12.443
13.240
14.848
15.659
16.473
16.473
16.473
16.473
16.473
17.292
18.939
19.768
19.768
20.599
21.434
22.271
23.952
23.952
 | 10.865
11.651
12.443
13.240
14.041
14.848
15.659
16.473
17.292
18.939
18.939
19.768
19.768
20.599
21.434
22.271
23.952
23.110
23.952
24.797 | 10.865
11.651
12.443
13.240
14.041
14.041
14.848
15.659
16.473
17.292
18.939
18.939
18.939
18.939
21.434
22.271
23.110
23.110
23.952
23.952
25.643 | 10.865
11.651
12.443
13.240
14.041
14.041
14.848
15.659
17.292
18.939
18.939
18.939
18.939
18.939
21.434
22.271
23.110
23.110
23.952
24.797
25.643
26.492
 | 10.865
11.651
12.443
13.240
14.041
14.041
14.848
15.659
17.292
18.939
19.768
18.939
19.768
21.434
22.271
22.271
22.271
22.271
22.271
22.271
22.3395
23.952
25.643
26.492
27.343 | 10.865
11.651
12.443
13.240
14.041
14.848
15.659
16.473
15.659
18.939
18.939
18.939
18.939
18.939
22.271
22.271
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.527
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.5272
22.52722
22.52722
22.52722
22.52722
22.52722
22.52722
22.52722
22.52722
22.52722
22.527222 |
10.865
11.651
12.443
13.240
14.041
14.041
14.041
14.041
14.041
14.848
17.292
18.939
19.768
19.768
19.797
22.271
23.952
23.110
23.110
23.952
23.110
23.952
23.110
23.952
23.110
23.952
23.110
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.952
23.110
23.952
23.952
23.952
23.110
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.052
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.051
23.0551
23.051
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.0551
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
23.05512
2 |
| 0.95 | 0.004 | 0.103 | 0.352 | 0.711 | 1.145 | 1.635 | 2.167 | 2.733 | 3.325 | 3.940 | 4.575 | 5.226 | 5.892 | 6.571 | 7.261 | 7.962 | 8.672 | | 9.390 | 9.390
10.117 | 9.390
10.117
10.851 | 9.390
10.117
10.851
11.591 | 9.390
10.117
10.851
11.591
12.338 | 9.390
10.117
10.851
11.591
12.338
13.091 | 9.390
10.117
10.851
11.591
12.338
13.091
13.848 | 9.390
10.117
10.851
11.591
11.591
12.338
13.091
13.848
13.848 | 9.390
10.117
10.851
11.591
12.338
12.338
13.091
13.848
14.611
15.379 | 9.390
10.117
10.851
11.591
12.338
12.338
13.091
13.848
13.848
14.611
15.379
16.151
 | 9.390
10.117
10.851
11.591
12.338
12.338
12.338
13.848
13.848
13.848
13.848
13.848
15.379
16.151 | 9.390
10.117
10.851
11.591
12.338
13.091
13.848
14.611
13.848
14.611
15.379
16.151
16.928
16.928 | 9.390
10.117
10.851
11.591
12.338
13.091
13.848
13.848
13.848
13.091
14.611
15.379
16.151
16.928
17.708
18.493
 | 9.390
10.117
10.851
11.591
12.338
13.091
13.848
13.848
13.848
13.091
15.379
16.928
16.928
16.928
17.708
18.493
19.281 | 9.390
10.117
10.851
11.591
12.338
12.338
12.338
13.091
13.848
13.091
15.379
15.379
16.928
16.928
16.928
17.708
18.493
19.281
19.281
 | 9.390
10.117
10.851
11.591
11.591
12.338
13.091
13.848
13.091
14.611
15.379
15.379
16.151
16.928
17.708
18.493
19.281
18.493
19.281
20.072
20.072 | 9.390
10.117
10.851
11.591
12.338
13.091
13.848
13.091
14.611
15.379
15.379
16.151
16.928
17.708
18.493
19.281
19.281
20.072
20.867
21.664
 | 9.390
10.117
10.851
11.591
11.591
12.338
13.091
13.848
13.091
15.379
16.151
16.928
16.928
17.708
18.493
19.281
19.281
20.072
20.072
21.664
21.664 | 9.390
10.117
10.851
11.591
12.338
13.091
13.848
13.091
15.379
16.928
16.928
16.928
17.708
19.281
19.281
19.281
20.072
20.867
21.664
22.465
23.269 | 9.390
10.117
10.851
11.591
12.338
12.338
12.338
13.091
15.379
16.151
16.928
16.928
16.928
17.708
18.493
19.281
19.281
19.281
20.072
20.867
21.664
22.465
23.269
23.269
 | 9.390
10.117
10.851
11.591
12.338
12.338
12.338
13.848
13.848
15.379
16.151
16.928
16.928
16.928
16.928
16.928
19.281
19.281
19.281
20.072
20.867
21.664
22.465
23.269
23.269
24.884 | 9.390
10.117
10.851
11.591
12.338
13.091
13.091
13.091
14.611
15.379
16.151
16.928
16.928
17.708
18.493
19.281
19.281
19.281
20.072
21.664
22.465
22.465
22.465
22.465
22.695 | 9.390
10.117
10.851
11.591
12.338
13.091
13.091
13.091
13.091
14.611
15.379
16.928
17.708
18.493
19.281
19.281
19.281
19.281
22.465
22.465
22.465
22.465
22.695
25.695
 |
| 0.975 | 0.001 | 0.051 | 0.216 | 0.484 | 0.831 | 1.237 | 1.690 | 2.180 | 2.700 | 3.247 | 3.816 | 4.404 | 5.009 | 5.629 | 6.262 | 6.908 | 7.564 | | 8.231 | 8.231
8.907 | 8.231
8.907
9.591 | 8.231
8.907
9.591
10.283 | 8.231
8.907
9.591
10.283
10.982 | 8.231
8.907
9.591
10.283
10.982
11.689 | 8.231
8.907
9.591
10.283
10.982
11.689
12.401 | 8.231
8.907
9.591
10.283
10.982
11.689
12.401
13.120 | 8.231
8.907
9.591
10.283
10.982
11.689
12.401
13.120
13.844 | 8.231
8.907
9.591
10.283
10.982
11.689
11.689
12.401
13.120
13.844
13.844
 | 8.231
8.907
9.591
10.283
10.982
11.689
11.689
11.689
13.120
13.120
13.844
14.573
15.308 | 8.231
8.907
9.591
10.283
10.982
11.689
12.401
13.120
13.844
13.844
13.844
13.844
15.308 | 8.231
8.907
9.591
10.283
10.982
11.689
12.401
13.120
13.844
13.844
13.844
13.844
13.844
14.573
15.308
16.047
 | 8.231
8.907
9.591
10.283
10.982
11.689
12.401
13.120
13.844
13.120
13.844
14.573
15.308
16.047
16.791
17.539 | 8.231
8.907
9.591
10.283
10.982
11.689
12.401
13.120
13.120
13.120
13.689
13.120
13.120
13.844
14.573
15.308
16.047
16.791
17.539
18.291
 | 8.231
8.907
9.591
10.283
10.982
11.689
12.401
13.120
13.844
14.573
15.308
15.308
16.791
16.791
16.791
17.539
18.291
19.047 | 8.231
8.907
9.591
10.283
10.283
11.689
12.401
13.120
13.844
13.120
13.844
14.573
15.308
16.791
16.791
16.791
16.791
16.791
18.291
19.047
19.806
 | 8.231
8.907
9.591
10.283
10.283
11.689
11.689
13.120
13.844
13.120
13.844
14.573
15.308
16.791
16.791
16.791
16.791
16.291
19.047
19.806
19.569 | 8.231
8.907
9.591
10.283
10.982
11.689
12.401
13.120
13.844
14.573
15.308
14.573
15.308
16.791
16.791
16.791
16.791
17.539
18.291
19.047
19.047
19.047
19.0569
20.569
21.336 | 8.231
9.591
9.591
10.283
10.982
11.689
11.689
13.120
13.120
13.844
14.573
15.308
15.308
15.308
16.791
17.539
16.791
17.539
18.291
19.047
19.066
20.569
21.336
22.106
 | 8.231
9.591
10.283
10.982
11.689
11.689
11.689
12.401
13.120
13.120
13.689
15.308
15.308
15.308
16.791
16.791
17.539
16.791
18.291
19.066
20.569
22.366
22.36878 | 8.231
8.907
9.591
10.283
10.283
10.982
11.689
12.401
13.120
13.120
13.844
14.573
14.573
15.308
14.573
15.308
16.791
16.791
16.791
15.308
16.791
19.047
19.291
19.205
20.569
22.336
22.336
22.336
22.336 | 8.231
8.907
9.591
10.283
10.283
11.689
11.689
12.401
13.120
13.844
14.573
15.308
15.308
16.791
16.791
16.791
16.791
16.791
16.791
16.291
19.067
19.806
20.569
21.336
22.106
22.878
23.654
23.654
 |
| 66.0 | 0.000 | 0.020 | 0.115 | 0.297 | 0.554 | 0.872 | 1.239 | 1.646 | 2.088 | 2.558 | 3.053 | 3.571 | 4.107 | 4.660 | 5.229 | 5.812 | 6.408 | | 7.015 | 7.015
7.633 | 7.015
7.633
8.260 | 7.015
7.633
8.260
8.897 | 7.015
7.633
8.260
8.897
9.542 | 7.015
7.633
8.260
8.897
9.542
10.196 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
11.524 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
11.524
11.524
12.198 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
11.524
11.524
12.198
12.879
 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
11.524
11.524
12.198
12.879
13.565 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
11.524
11.524
11.524
12.198
12.879
13.565
13.565 | 7.015
7.633
8.260
8.897
9.542
10.196
11.524
11.524
11.524
12.856
11.526
11.526
11.526
11.256
 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
10.856
11.524
12.198
12.879
12.879
13.565
14.256
14.953
15.655 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
11.524
12.198
12.198
12.198
12.879
12.879
12.879
12.879
12.879
14.953
14.256
14.953
15.655
 | 7.015
7.633
8.260
8.897
9.542
10.196
11.524
11.524
12.879
13.565
14.953
13.565
14.953
13.655
14.953
15.655
16.362
17.074 | 7.015
7.633
8.260
8.897
9.542
10.196
11.524
11.524
11.524
12.856
11.526
14.953
13.565
14.953
14.953
15.655
14.953
15.655
14.953
17.074
17.074
 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
11.524
11.524
12.879
12.879
12.879
12.879
13.565
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
14.256
11.256
14.256
11.256
11.256
11.256
11.256
11.256
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.2566
11.25666
11.25666
11.25666
11.25666
11.25666
11.25666
11.25666
11.25666
11.25666
11.256666
11.2566666666666666666666666666666666666 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
11.524
12.198
12.879
12.879
12.879
12.879
12.879
13.565
14.256
14.256
14.256
14.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
114.256
115.267
115.267
115.267
115.267
115.267
116.256
116.256
117.256
116.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.256
117.2566
117.256
117.2566
117.2566
117.2566
117.2566
117.2566 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
11.524
12.198
12.198
12.198
12.198
12.879
13.565
14.953
14.953
15.655
14.953
15.655
14.953
15.655
14.953
15.655
119.233
19.233
 | 7.015
7.633
8.260
8.897
9.542
10.196
11.524
11.524
11.524
12.879
13.565
14.953
14.953
14.953
14.953
14.953
14.953
13.565
14.953
13.565
14.953
13.565
11.789
13.565
11.789
19.233
19.233
19.233 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
11.524
11.524
12.879
13.565
14.953
14.953
14.953
14.953
14.953
14.953
14.256
14.256
14.256
14.256
14.256
19.333
17.789
19.233
19.260
20.691 | 7.015
7.633
8.260
8.897
9.542
10.196
10.856
11.524
11.524
12.879
12.879
13.565
14.953
14.953
14.953
14.953
15.655
14.953
17.074
17.789
18.509
18.509
19.233
19.233
19.233
20.691
22.164
 |
| 0.995 | 0.000 | 0.010 | 0.072 | 0.207 | 0.412 | 0.676 | 0.989 | 1.344 | 1.735 | 2.156 | 2.603 | 3.074 | 3.565 | 4.075 | 4.601 | 5.142 | 5.697 | | 6.265 | 6.265
6.844 | 6.265
6.844
7.434 | 6.265
6.844
7.434
8.034 | 6.265
6.844
7.434
8.034
8.643 | 6.265
6.844
7.434
8.034
8.643
9.260 | 6.265
6.844
7.434
8.034
8.643
9.260
9.260 | 6.265
6.844
7.434
8.034
8.643
9.260
9.886
9.886 | 6.265
6.844
7.434
8.034
8.643
9.260
9.260
9.886
10.520
11.160 | 6.265
6.844
7.434
8.034
8.643
9.260
9.260
9.886
10.520
11.160
11.808
 | 6.265
6.844
7.434
8.034
8.643
9.260
9.260
9.886
10.520
11.160
11.808
12.461 | 6.265
6.844
7.434
8.034
8.643
9.260
9.260
9.886
10.520
11.160
11.808
12.461
12.461 | 6.265
6.844
7.434
8.034
8.643
9.260
9.260
9.886
10.520
11.160
11.160
11.808
11.808
11.808
11.3.787
 | 6.265
6.844
7.434
8.034
8.643
9.260
9.260
9.260
11.160
11.160
11.808
11.808
11.808
11.808
11.461
13.787
13.787
13.787 | 6.265
6.844
7.434
8.034
8.643
9.260
9.260
9.886
10.520
11.160
11.808
11.160
11.808
11.160
11.808
11.160
11.461
13.787
14.458
15.134
 | 6.265
6.844
7.434
8.034
8.643
9.260
9.886
10.520
11.160
11.808
11.160
11.808
11.461
13.787
13.787
13.787
13.787
15.134
15.815 | 6.265
6.844
7.434
8.034
8.643
9.260
9.260
10.520
11.160
11.160
11.160
11.808
12.461
13.787
13.787
13.787
13.787
13.787
15.134
15.815
16.501
 | 6.265
6.844
7.434
8.034
8.643
9.260
9.260
9.260
11.160
11.160
11.160
11.808
12.461
13.121
13.787
13.787
13.787
13.787
13.787
13.121
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.787
13.777
13.7777
13.77777
13.77777777777 | 6.265
6.844
7.434
8.034
8.643
9.260
9.260
9.886
10.520
11.160
11.160
11.160
11.160
11.458
12.461
13.787
14.458
12.121
13.787
15.815
15.815
17.192
17.192 | 6.265 6.844 7.434 8.034 8.643 9.260 9.886 11.160 11.160 11.160 11.160 11.160 11.160 11.151 12.451 12.458 15.815 16.501 17.192 17.192 18.586 18.586
 | 6.265 6.844 7.434 8.034 8.643 9.260 9.886 10.520 11.160 11.808 11.808 11.160 11.808 11.15.121 11.15.121 11.2461 11.2461 11.458 15.815 15.815 15.815 15.815 15.815 16.501 17.887 192 192 192 192 192 192 192 192 192 15.815 15.815 16.501 17.887 192.289 19.289 | 6.265 6.844 7.434 8.034 8.643 9.260 9.886 10.520 11.160 11.160 11.2461 12.461 13.787 13.787 14.458 12.458 15.815 12.461 13.787 14.458 15.815 16.501 17.192 19.289 19.289 19.289 | 6.265 6.265 6.844 8.634 8.643 9.260 9.886 10.520 11.160 11.160 |
| grees of
eedom | 1 | 7 | С | 4 | ß | 9 | 7 | ω | 6 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | α |)
1 | 16 | 20 10 | 19
20
21 | 2 1 0 6 7 7 0
2 7 0 8 7 7 0
2 7 0 8 | 19
20
22
22
23 | 1 0 0 1 0 0 7 0 0 7 0 0 7 0 0 7 0 0 7 0 0 7 0 0 7 0 | 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | 2 2 3 7 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 2 2 3 3 5 4 7 3 5 5 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | 2010
2010
2010
2010
2010
2010
2010
2010
 | 3 3 5 1 0 0 8 3 4 9 2 7 9 6 2 7 8 3 7 1 0 0 8 3 7 9 7 8 5 7 8 7 8 7 9 7 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 | 4 3 3 5 1 0 0 8 4 9 2 1 0 6 7 4 3 5 1 0 0 8 4 8 5 7 8 5 7 8 5 7 8 7 8 7 9 7 7 8 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 8 7 8 7 8 8 7
8 7 | D | 0 2 4 3 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 | 7 0 0 7 7 9 7 0 0 0 0 7 0 0 7 0 0 7 0 0 0 0
 | 8 7 9 2 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 | 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 |

 $1 - \alpha$