Hellenic Complex Systems Laboratory

Quality Control Using Convolutional Neural Networks Applied to Samples of Very Small Size

Technical Report XXI

Rallou A. Chatzimichail and Aristides T. Hatjimihail
2022

Quality Control Using Convolutional Neural Networks Applied to Samples of Very Small Size

Rallou A. Chatzimichail, PhD [a], Aristides T. Hatjimihail, MD, PhD [a]

[a] *Hellenic Complex Systems Laboratory*

Search Terms: QC, convolutional neural networks, probability for rejection, probability for false rejection, QC measurement

1. Abstract

Although there is extensive literature on the application of artificial neural networks in quality control (QC), to monitor the conformity of a process to quality specifications, at least five QC measurements are required, increasing the related cost. To explore the application of neural networks to samples of QC measurements of very small size, four one-dimensional (1-D) convolutional neural networks (CNN) were designed, trained, and tested with datasets of n-tuples of simulated standardized normally distributed QC measurements, for $1 \leq n \leq 4$.

The designed neural networks were compared to statistical QC functions with equal probabilities for false rejection, applied to samples of the same size. When the n-tuples included at least two QC measurements distributed as $\mathcal{N}(\mu, \sigma^2)$, where $0.2 < |\mu| \leq 6.0$, and $1.0 < \sigma \leq 7.0$, the designed neural networks outperformed the respective statistical QC functions. Therefore, 1-D CNN applied to samples of 2-4 quality control measurements can be used to increase the probability of detection of the nonconformity of a process to the quality specifications, with lower cost.

2. Introduction

Alternative quality control (QC) procedures can be applied to a process to test the null hypothesis, that the process conforms to the quality specifications and consequently is in control, against the alternative, that the process is out of control. When a true null hypothesis is rejected, a statistical type I error is committed. We have then a false rejection of a run of the process. The probability of a type I error is called probability for false rejection. When a false null hypothesis is accepted, a statistical type II error is committed. We fail then to detect a significant change of the probability density function of a quality characteristic of the process. The probability for rejection of a false null hypothesis equals the probability of detection of the nonconformity of the process to the quality specifications. A QC procedure can be formulated as a Boolean expression of QC functions, applied to a sample of QC measurements. If it is true, then the null hypothesis is considered as false, the process as out of control, and the run is rejected [1].

A statistical QC procedure is evaluated as a Boolean proposition of one or more statistical QC functions. Each QC function is a decision rule, evaluated by calculating a statistic of the measured quality characteristic of a sample of QC measurements. Then, if the statistic is out of the interval between the decision limits, the decision rule is considered as true. If it is true, then the null hypothesis is considered as false, the process as out of control, and the run is rejected. Control charts are plots of the statistics, in time order.

Artificial neural networks (ANN) are adaptive computational algorithms, for statistical data modeling and classification of arbitrary precision, inspired by the brain structure and information processing. They consist of interconnected input, processing, and output nodes, ordered in input, hidden and output layers respectively. Information flows from input to output, through weighed directed interconnections. Each processing node receives as input the sum of the weighed outputs of other nodes plus a bias. The output of each node is calculated by an activation function and is sent to other nodes. Applying repeatedly learning rules, the weights and biases are adapted to minimize an output dependent loss function. ANN are equivalent to nonlinear mathematical functions $\mathbf{w} = \mathrm{F}(\mathbf{Q})$, where $\mathbf{w}$ and $\mathbf{Q}$ are vectors, matrices, or tensors of finite dimensions.

Since 1943 [2], there have been designed many ANN architectures, growing in complexity [3], evolving into multilayer deep neural networks (DNN). Convolutional neural networks (CNN) are a type of DNN, used extensively for feature extraction in image processing [4]. One-dimensional (1-D) CNN have been used in signal processing and classification, including anomaly detection in quality control [5]. As they combine both feature extraction and classification tasks into a single structure, they are compact with low computational complexity. Therefore, they are easier to implement in relatively low-cost systems. CNN are trained and then tested using appropriate datasets to estimate the correct classification and misclassification rates.

ANN are bioinspired computational algorithms, as they are the genetic algorithms (GA). Since 1989, ANN have been proposed as alternative to statistical quality control procedures [6] while GA have been used for the design of QC since 1993 [7].

The first application of ANN to QC was published by Pugh [8], who applied ANN to samples of 5 QC measurements, with performance comparable to that of simple statistical quality control rules. Hwarng and Hubele [9] applied ANN to samples of 8 QC measurements for control charts pattern recognition among 8 unnatural control chart patterns. Other early applications of ANN to QC were published by Smith [10], Stützle [11], Cheng [12] and Chang and Aw [13]. The application of ANN to SPC was reviewed comprehensively by Psarakis [14]. Hachicha and Ghorbel [15] reviewed extensively their application to control charts pattern recognition.

The input of the ANN applied to SPC was either n - tuples of QC measurements or derived statistics. To the best of our knowledge there have not been any publication in quality control about ANN applied to n - tuples of QC measurements for $n < 5$. Therefore, to explore the application of ANN to samples of QC measurements of very small size, four 1-D CNN were designed and applied to n - tuples of QC measurements, for $1 \leq n \leq 4$, as described in detail in the *Methods* section.

3. Methods

3.1. Simulated QC Sample Tuples

It is assumed that u are standardized normally distributed measurements of control samples when the measurement process is in control and v when it is out of control. Then:

$$U \sim \mathcal{N}(0,1)$$

$$V \sim \mathcal{N}(\mu, \sigma^2)$$

where:

$$|\mu| > 0 \vee \sigma > 1$$

Series of QC measurements n-tuples (QCMT) were simulated, with up to *k* measurements out of control, in all possible combinations, where:

$$1 \leq n \leq 4$$

$$1 \leq k \leq n$$

The combinations of the elements of the QCMT are presented in Table 1.

Table 1

The possible control sample measurements tuples (QCMT)

			k			
			1	2	3	4
n	1	$[u_1]$	$[v_1]$			
	2	$[u_1, u_2]$	$[u_1, v_1]$ $[v_1, u_1]$	$[v_1, v_2]$		
	3	$[u_1, u_2, u_3]$	$[u_1, u_2, v_1]$ $[u_1, v_1, u_2]$ $[v_1, u_1, u_2]$	$[u_1, v_1, v_2]$ $[v_1, u_1, v_2]$ $[v_1, v_2, u_1]$	$[v_1, v_2, v_3]$	
	4	$[u_1, u_2, u_3, u_4]$	$[u_1, u_2, u_3, v_1]$ $[u_1, u_2, v_1, u_3]$ $[u_1, v_1, u_2, u_3]$ $[v_1, u_1, u_2, u_3]$	$[u_1, u_2, v_1, v_2]$ $[u_1, v_1, u_2, v_2]$ $[u_1, v_1, v_2, u_2]$ $[v_1, u_1, u_2, v_2]$ $[v_1, u_1, v_2, u_2]$ $[v_1, v_2, u_1, u_2]$	$[u_1, v_1, v_2, v_3]$ $[v_1, u_1, v_2, v_3]$ $[v_1, v_2, u_1, v_3]$ $[v_1, v_2, v_3, u_1]$	$[v_1, v_2, v_3, v_1]$

For $1 \leq n \leq 4$, *assuming* $\mathrm{U} \sim \mathcal{N}(0,1)$ *and* $\mathrm{V} \sim \mathcal{N}(\mu, \sigma^2)$

3.2.Simulated datasets

Training and testing datasets were created with simulated QC measurements, normally distributed and standardized.

3.2.1. Training Datasets

For $a \in \{4,6,8\}$, and $1 \leq n \leq 4$, each training dataset $\mathbf{T}_a(n)$ consists of:

1. $a\, 2^{n-1} 10^5$ n-tuples of random simulated QC measurements distributed as $\mathcal{N}(0,1)$,
2. $2^{n-1} 10^5$ n-tuples of random simulated QC measurements, k distributed as $\mathcal{N}(0, \sigma^2)$ and $n - k$ *as* $\mathcal{N}(0,1)$, for $1 \leq \mathrm{k} \leq \mathrm{n}$ and $1 < \sigma \leq 11$.
3. $2^{n-1}\, 10^5$ n-tuples of random simulated QC measurements, k distributed as $\mathcal{N}(\mu, 1)$ and $n - k$ *as* $\mathcal{N}(0,1)$, for $1 \leq \mathrm{k} \leq \mathrm{n}$ and $0 < |\mu| \leq 10$, and

3.2.2. Testing Datasets

The following testing datasets were created:

1. $\mathbf{D}(n, 0, 1)$, for $1 \leq n \leq 4$, with 10^8 n-tuples of random simulated QC measurements distributed as $\mathcal{N}(0,1)$.
2. $\mathbf{D}(n, k, 0, \sigma)$ for $\sigma = 1.1, 1.2, \ldots, 7.0$, and $1 \leq n \leq 4$ and $1 \leq k \leq n$, with $\binom{n}{k} 10^5$ n-tuples of random simulated QC measurements, k distributed as $\mathcal{N}(0, \sigma^2)$ and $n - k$ *as* $\mathcal{N}(0,1)$.
3. $\mathbf{D}(n, k, \mu, 1)$ for $|\mu| = 0.1, 0.2, \ldots, 6.0$, and $1 \leq n \leq 4$ and $1 \leq k \leq n$, with $\binom{n}{k} 10^5$ n-tuples of random simulated QC measurements, k distributed as $\mathcal{N}(\mu, 1)$ and $n - k$ *as* $\mathcal{N}(0,1)$

Tables 2 and 3 present the elements of the simulated training and testing datasets, with the respective numbers of the QCMT.

Table 2

The simulated training datasets, for $\alpha \in \{4,6,8\}$

		n	U	V
$\mathbf{T}_a(1)$	$\{[u_1], False\}_{i=1}^{n}$	$a\ 10^5$	$\mathcal{N}(0,1)$	
	$\{[v_1], True\}_{i=1}^{n}$	10^5		$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5		$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
$\mathbf{T}_a(2)$	$\{[u_1,u_2], False\}_{i=1}^{n}$	$3a\ 10^5$	$\mathcal{N}(0,1)$	
	$\{[u_1,v_1], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,u_1], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,v_2], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
$\mathbf{T}_a(3)$	$\{[u_1,u_2,u_3], False\}_{i=1}^{n}$	$7a\ 10^5$	$\mathcal{N}(0,1)$	
	$\{[u_1,u_2,v_1], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[u_1,v_1,u_2], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,u_1,u_2], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[u_1,v_1,v_2], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,u_1,v_2], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,v_2\ ,u_1], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathrm{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,v_2,v_3], True\}_{i=1}^{n}$	10^5		$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5		$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
$\mathbf{T}_a(4)$	$\{[u_1,u_2,u_3,u_4], False\}_{i=1}^{n}$	$1.5a\ 10^6$	$\mathcal{N}(0,1)$	
	$\{[u_1,u_2,u_3,v_1], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[u_1,u_2,v_1,u_3], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[u_1,v_1,u_2,u_3], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,u_1,u_2,u_3], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[u_1,u_2,v_1,v_2], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[u_1,v_1,u_2,v_2], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[u_1,v_1,v_2,u_2], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,u_1,u_2,v_2], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,u_1,v_2,u_2], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,v_2,u_1,u_2], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[u_1,v_1,v_2,v_3], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,u_1,v_2,v_3], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,v_2,u_1,v_3], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,v_2,v_3,u_1], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$
	$\{[v_1,v_2,v_3,v_4], True\}_{i=1}^{n}$	10^5		$\mathcal{N}(0,\sigma^2), 1 < \sigma \leq 11$
		10^5		$\mathcal{N}(\mu,1), 0 < \lvert\mu\rvert \leq 10$

Table 3
The simulated testing datasets

		n	U	V
D(1,0,1)	$\{[u_1], False\}_{i=1}^{n}$	8 10^5	$\mathcal{N}(0,1)$	
D(1,1,μ,σ)	$\{[v_1], True\}_{i=1}^{n}$	10^5		$\mathcal{N}(\mu,\sigma^2), \lvert\mu\rvert > 0 \vee \sigma > 1$
D(2,0,1)	$\{[u_1, u_2], False\}_{i=1}^{n}$	8 10^5	$\mathcal{N}(0,1)$	
D(2,1,μ,σ)	$\{[u_1, v_1], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\sigma^2), \lvert\mu\rvert > 0 \vee \sigma > 1$
	$\{[v_1, u_1], True\}_{i=1}^{n}$	10^5		
D(2,2,μ,σ)	$\{[v_1, v_2], True\}_{i=1}^{n}$	10^5		$\mathcal{N}(\mu,\sigma^2), \lvert\mu\rvert > 0 \vee \sigma > 1$
D(3,0,1)	$\{[u_1, u_2, u_3], False\}_{i=1}^{n}$	8 10^5	$\mathcal{N}(0,1)$	
D(3,1,μ,σ)	$\{[u_1, u_2, v_1], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\sigma^2), \lvert\mu\rvert > 0 \vee \sigma > 1$
	$\{[u_1, v_1, u_2], True\}_{i=1}^{n}$	10^5		
	$\{[u_1, u_2, v_1], True\}_{i=1}^{n}$	10^5		
D(3,2,μ,σ)	$\{[u_1, v_1, v_2], True\}_{i=1}^{n}$	10^5		
	$\{[v_1, u_1, v_2], True\}_{i=1}^{n}$	10^5		
	$\{[v_1, v_2, u_1], True\}_{i=1}^{n}$	10^5		
D(3,3,μ,σ)	$\{[v_1, v_2, v_3], True\}_{i=1}^{n}$	10^5		$\mathcal{N}(\mu,\sigma^2), \lvert\mu\rvert > 0 \vee \sigma > 1$
D(4,0,1)	$\{[u_1, u_2, u_3, u_4], False\}_{i=1}^{n}$	8 10^5	$\mathcal{N}(0,1)$	
D(4,1,μ,σ)	$\{[u_1, u_2, u_3, v_1], True\}_{i=1}^{n}$	10^5	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\sigma^2), \lvert\mu\rvert > 0 \vee \sigma > 1$
	$\{[u_1, u_2, v_1, u_3], True\}_{i=1}^{n}$	10^5		
	$\{[u_1, v_1, u_2, u_3], True\}_{i=1}^{n}$	10^5		
	$\{[v_1, u_1, u_2, u_3], True\}_{i=1}^{n}$	10^5		
D(4,2,μ,σ)	$\{[u_1, u_2, v_1, v_2], True\}_{i=1}^{n}$	10^5		
	$\{[u_1, v_1, u_2, v_2], True\}_{i=1}^{n}$	10^5		
	$\{[u_1, v_1, v_2, u_2], True\}_{i=1}^{n}$	10^5		
	$\{[v_1, u_1, u_2, v_2], True\}_{i=1}^{n}$	10^5		
	$\{[v_1, u_1, v_2, u_2], True\}_{i=1}^{n}$	10^5		
	$\{[v_1, v_2, u_1, u_2], True\}_{i=1}^{n}$	10^5		
D(4,3,μ,σ)	$\{[u_1, v_1, v_2, v_3], True\}_{i=1}^{n}$	10^5		
	$\{[v_1, u_1, v_2, v_3], True\}_{i=1}^{n}$	10^5		
	$\{[v_1, v_2, u_1, v_3], True\}_{i=1}^{n}$	10^5		
	$\{[v_1, v_2, v_3, u_1], True\}_{i=1}^{n}$	10^5		
D(4,4,μ,σ)	$\{[v_1, v_2, v_3, v_4], True\}_{i=1}^{n}$	10^5		$\mathcal{N}(\mu,\sigma^2), \lvert\mu\rvert > 0 \vee \sigma > 1$

3.3. QC functions

As QC function we define a Boolean valued function applied to a n-tuple $\boldsymbol{x}$ of QC measurements of a run of a process. If it is true, the process is considered out of control, and the run is rejected.

3.3.1. Neural Network QC functions

As $N_\alpha(\boldsymbol{x})$ is denoted a neural network QC function. For this project, there were designed four 1-D CNN with 69 layers, 486 – 11634 nodes and 1x1 kernels. Fig. 1 presents their template. In Table 4 there is a summary presentation of their layers and nodes.

Table 4
CNN Layers and nodes

CNN	$N_a(x_1)$		$N_a(x_1, x_2)$		$N_a(x_1, x_2, x_3)$		$N_a(x_1, x_2, x_3, x_4)$	
	Layers	Nodes	Layers	Nodes	Layers	Nodes	Layers	Nodes
Input	1	1	1	2	1	3	1	4
Hidden	67	483	67	2102	67	5577	67	11628
Output	1	2	1	2	1	2	1	2

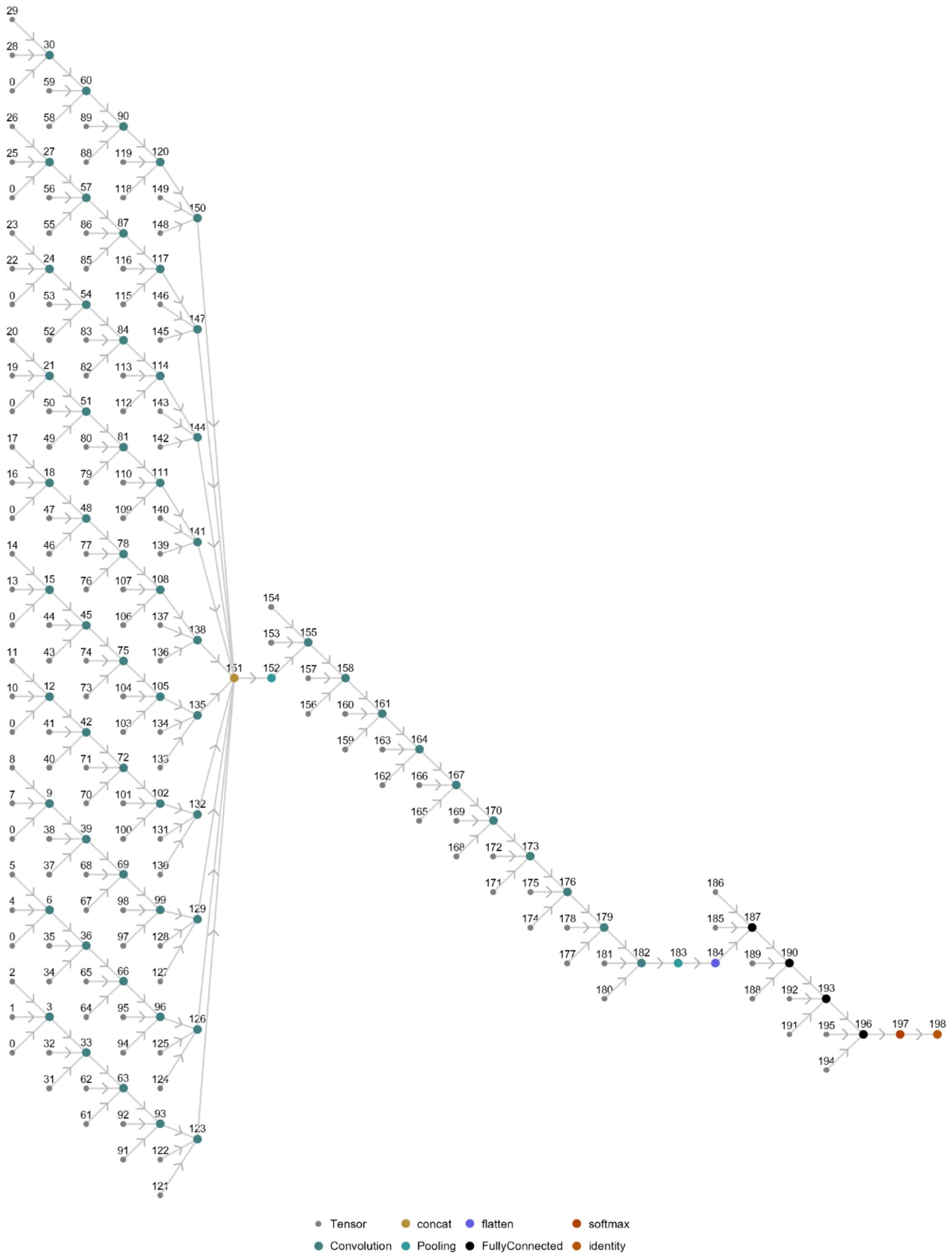

Fig. 1 The template of the CNN $N_a(\boldsymbol{x})$

For $a \in \{4,6,8\}$, $n = \#\boldsymbol{x}$ *(that is n equal to the cardinality of* $\boldsymbol{x}$*) and* $1 \leq n \leq 4$, each 1-D CNN $N_\alpha(\boldsymbol{x})$ consists of:

1. The input layer, with n nodes.
2. Ten parallel arrays of five convolutional layers each. Every convolutional layer has n output channels, and kernels of size *1x1.*

1. A catenating layer, which catenates the ten parallel arrays.
2. A pooling layer, performing 1-D pooling with kernels of size *2x1.*

3. Ten convolutional net layers having n output channels, and kernels of size *1x1.*
4. A pooling net layer, performing 1-D pooling with kernels of size *2x1.*
5. *Four linear net layers with output vectors of decreasing size.*
6. *A softmax net layer, n*ormalizing the exponential of the output vector of the fourth linear layer.
7. *The output layer, which is a decoder with two nodes.*

For $a \in \{4,6,8\}$, each CNN $N_\alpha(\boldsymbol{x})$ was trained using the respective training set $\mathbf{T}_\alpha(n)$, and then tested on the datasets $\mathbf{D}(n, 0,1)$ and $\mathbf{D}(n, k, \mu, \sigma)$, for each combination of k and n, for $1 \leq n \leq 4$ and $1 \leq k \leq n$. The misclassification rate of the application of $N_\alpha(\boldsymbol{x})$ to the dataset $\mathbf{D}(n, 0,1)$ equals its probability for false rejection, denoted as $P_{N,a}(n, \mu, \sigma)$. The correct classification rate of the application of $N_\alpha(\boldsymbol{x})$ to the dataset $\mathbf{D}(n, k, \mu, \sigma)$, for $|\mu| > 0$ and $\sigma > 1$, equals its probability for rejection, denoted as $P_{N,a}(n, k, \mu, \sigma)$ (see *Notation and Formalism*).

The twelve trained CNN are available at https://www.hcsl.com/Supplements/CNN.zip, in *Apache MXNet, ONNX, and Wolfram Language frameworks formats, as described* in *Supplement 1.*

3.3.2. Statistical QC Functions

Let m the expected mean and s the standard deviation of the QC measurements x_i of a measurement process when the process is in control. As $S_\mathrm{a}(\boldsymbol{x}; l, \mathrm{m}, \mathrm{s})$ is denoted a statistical QC function, applied to a tuple $\boldsymbol{x}$ of QC measurements x_i, which is true if $|x_i - m| \geq l\ s$, for any of them.

For $\alpha = 4, 6, 8$ the probability for rejection of the application of $S_\mathrm{a}(\boldsymbol{x}; l, 0,1)$ to the dataset $D(n, 0,1)$ is denoted as $P_{S,a}(n, 0,1)$. *The decision limit l of* $S_\mathrm{a}(\boldsymbol{x}; l, 0,1)$ *is defined* (see Formalism and Notation) *so that:*

$$P_{S,a}(n, 0,1) \ = \ P_{N,a}(n, 0,1)$$

For $a \in \{4,6,8\}$, and each combination of k and n, and $1 \leq n \leq 4$ and $1 \leq k \leq n$, the probability for rejection of the application of $S_\mathrm{a}(\boldsymbol{x}; l, 0,1)$ to the dataset $D(n, k, \mu, \sigma)$ is denoted as $P_{S,a}(n, k, \mu, \sigma)$.

The calculated values of l are presented in Table 5.

Table 5

The probabilities for false rejection $P_{N,a}(n, 0,1)$ and the calculated values of the decision limits l

CNN	n	α	$P_{N,a}(n, 0,1)$	l
$N_a(x_1)$	1	4	0.023750	2.261149
		6	0.012258	2.504643
		8	0.008197	2.643835
$N_a(x_1, x_2)$	2	4	0.008733	2.849716
		6	0.016204	2.646416
		8	0.009049	2.838333
$N_a(x_1, x_2,\ x_3)$	3	4	0.007836	3.009274
		6	0.009234	2.958895
		8	0.005146	3.135029
$N_a(x_1, x_2,\ x_3, x_4)$	4	4	0.004909	3.231921
		6	0.010381	3.010814
		8	0.005599	3.194108

3.4.Computer System and Software

The datasets were created and the CNN were designed, trained, and tested using Wolfram Mathematica Ver. 13.0, Wolfram Research, Inc., Champaign, IL, USA, on a computer system with an Intel Core i9-11900K CPU, 128 GB RAM, a NVIDIA GeForce RTX 30380 Ti GPU, under Microsoft Windows 11 Professional OS.

4 Results

The plots of the probabilities for rejection $P_{N,a}(n,k,\mu,\sigma)$ and $P_{S,a}(n,k,\mu,\sigma)$ of the application of the QC functions to the respective datasets, their differences $\Delta P_a(n,k,\mu,\sigma)$, and their relative differences $\Delta P_{R,a}(n,k,\mu,\sigma)$ are presented in *Appendix*. The probabilities for rejection $P_{N,a}(n,k,\mu,\sigma)$ and $P_{S,a}(n,k,\mu,\sigma)$, their differences $\Delta P_a(n,k,\mu,\sigma)$ with the respective statistical significance (p-values), and their relative differences $\Delta P_{R,a}(n,k,\mu,\sigma)$ are presented in detail in *Supplement 2*.

As examples, Fig 2 and 3 present the probabilities for rejection $P_{N,a}(2,k,\mu,\sigma)$ of the application of the CNN functions to the respective datasets.

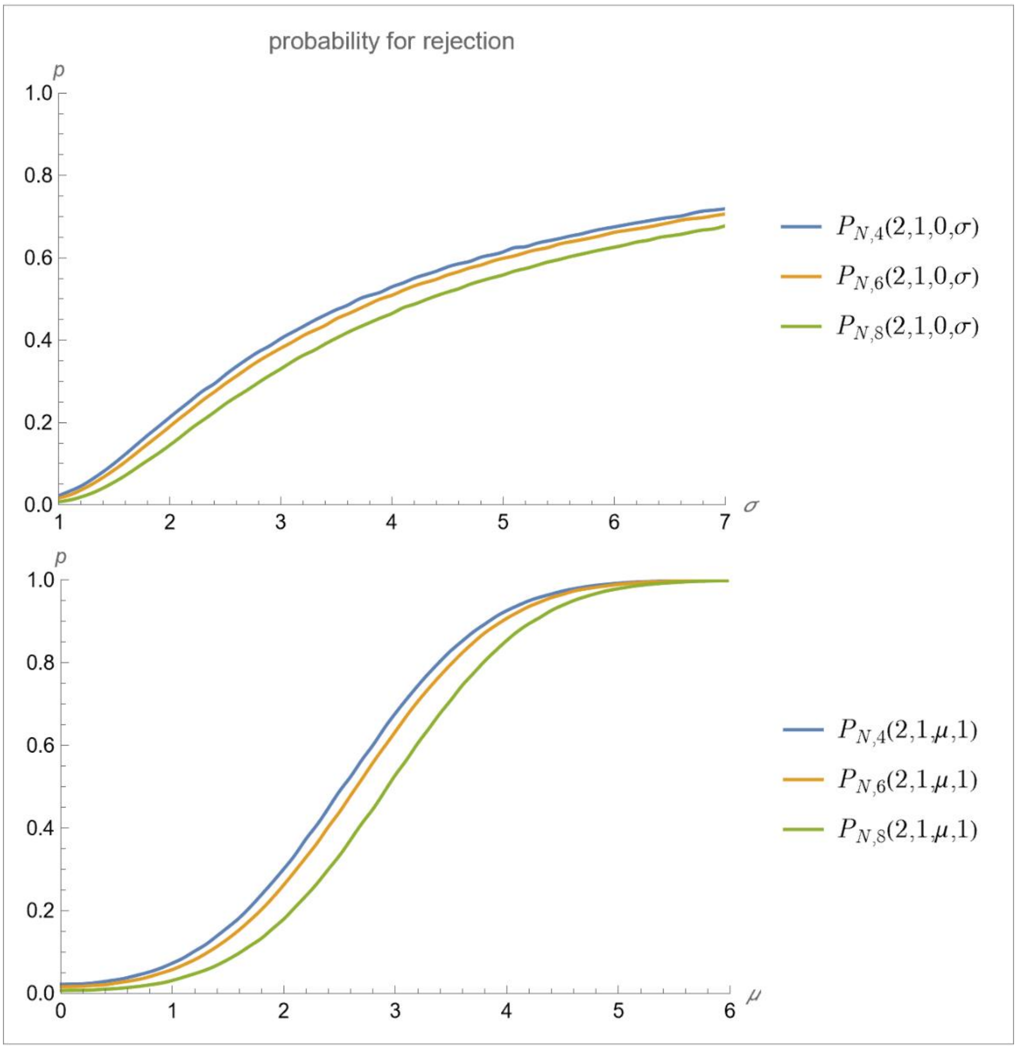

Fig 2 The probabilities for rejection of the CNN $N_4(x_1,x_2), N_6(x_1,x_2), N_8(x_1,x_2)$ when applied to two QC measurements, one distributed as $\mathcal{N}(0,1)$, and one as $\mathcal{N}(0,\sigma^2)$ *(upper plot) or as* $\mathcal{N}(\mu,1)$ *(lower plot)*

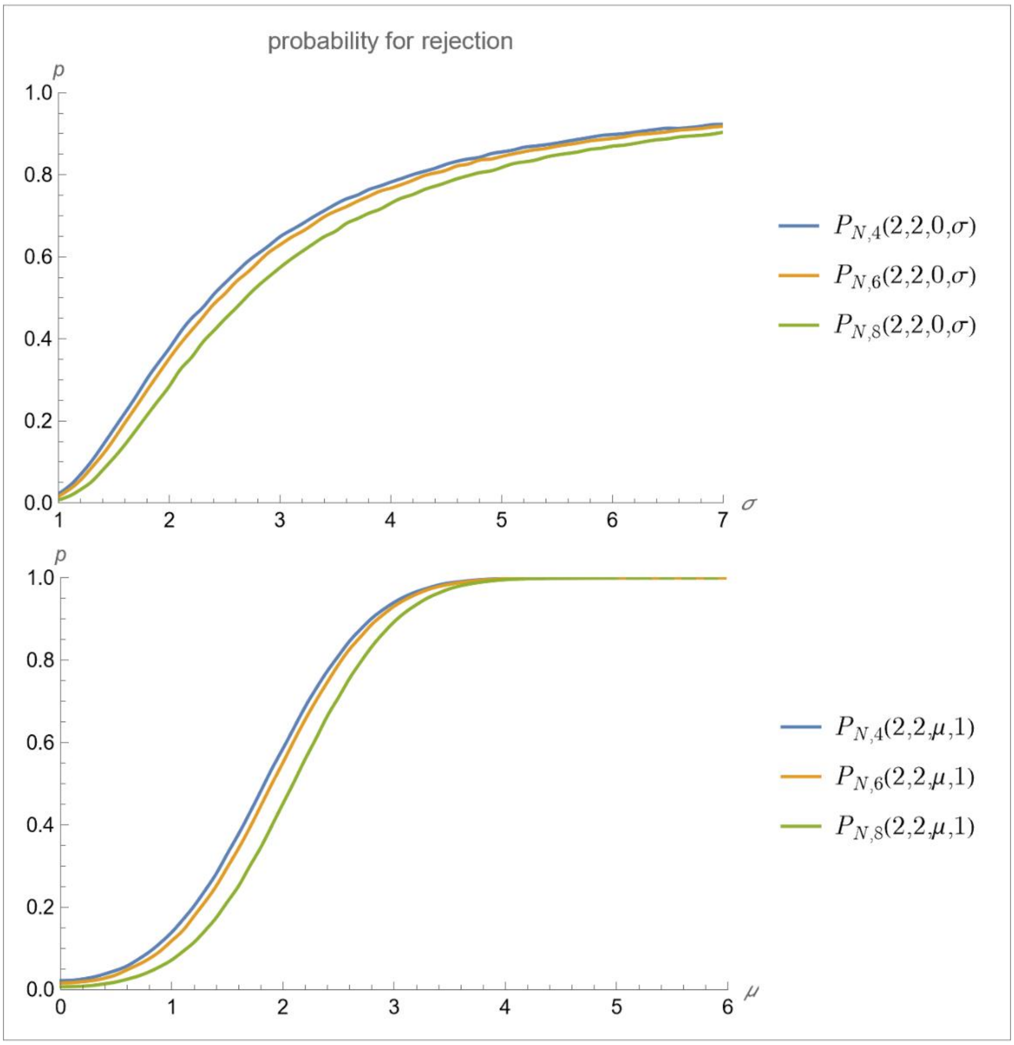

Fig 3 The probabilities for rejection of the CNN $N_4(x_1, x_2), N_6(x_1, x_2), N_8(x_1, x_2)$ when applied to two QC measurements distributed as $\mathcal{N}(0, \sigma^2)$ *(upper plot) or as* $\mathcal{N}(\mu, 1)$ *(lower plot)*

Likewise, Fig 4 and 5 present the differences $\Delta \mathrm{P}_8(n, k, \mu, \sigma)$ of the probabilities for rejection $P_{N,8}(n, k, \mu, \sigma)$ and $P_{S,8}(n, k, \mu, \sigma)$ of the application of the CNN and statistical QC functions to the respective datasets.

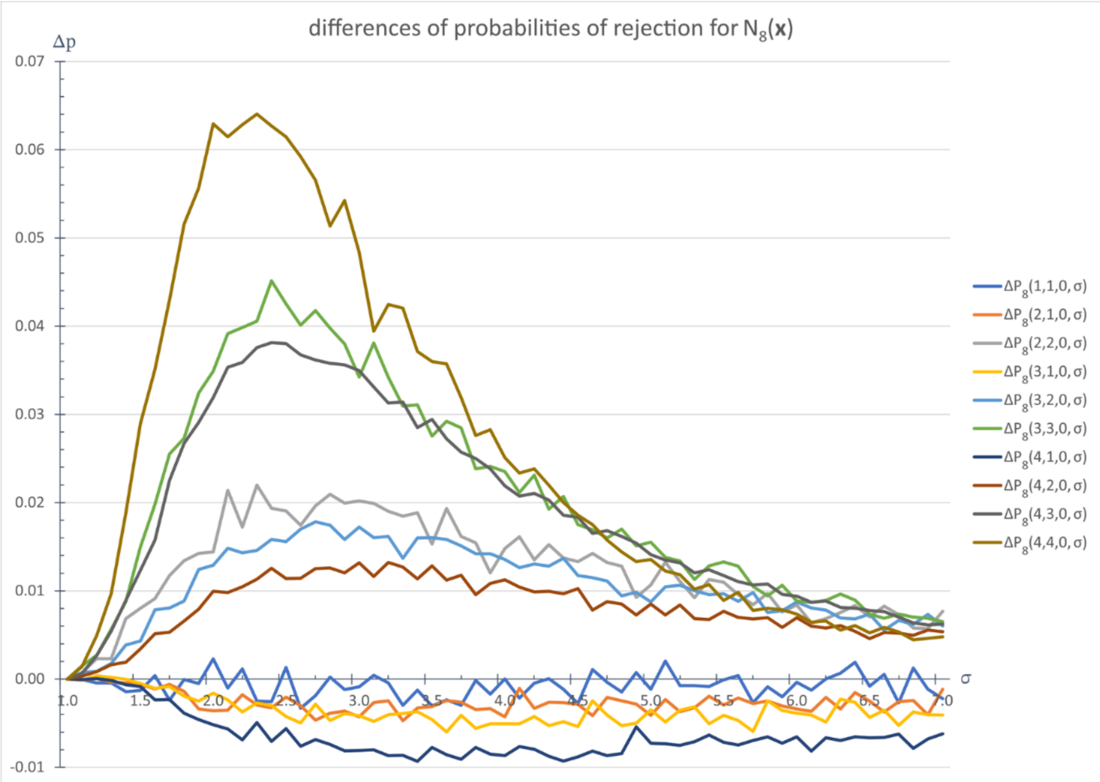

Fig 4 The differences between the probabilities for rejection of $N_8(\boldsymbol{x})$ and the respective statistical QC functions vs σ

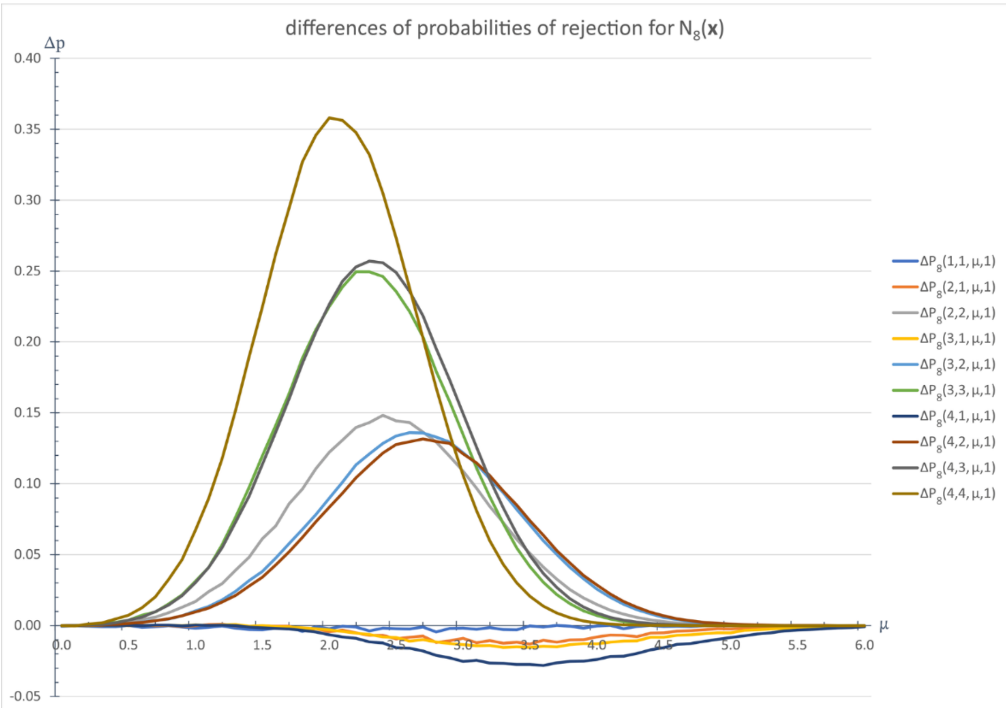

Fig 5 The relative differences between the probabilities for rejection of $N_8(\boldsymbol{x})$ and the respective statistical QC functions vs μ

The results show that for $\alpha = 4, 6, 8$, and $1 < n \leq 4$, $1 < k \leq n$, $0.2 < |\mu| \leq 6.0$, and $1.0 \leq \sigma \leq 7.0$

$$\Delta P_a(n, k, \mu, \sigma) > 0$$

For $a \in \{4,6,8\}$, $1 < n \leq 4$, $1 < k \leq n$, $0.2 < |\mu| \leq 6.0$, and $1.2 < \sigma \leq 7.0$, the differences are statistically significant ($p < 0.01$), except for some differences $\Delta P_a(n,k,\mu,\sigma) < 0.002$, with $P_{N,a}(n,k,\mu,\sigma) > 0.995$ and $P_{S,a}(n,k,\mu,\sigma) > 0.995$ (see *Supplement 2*).

For $a \in \{4,6,8\}$, and $1 < n \leq 4$, most of the differences $\Delta P_a(n,1,\mu,\sigma)$ are negative, with $-0.03 < \Delta P_a(n,1,\mu,\sigma)$ (see *Supplement 2*).

For $a \in \{4,6,8\}$, and $1 < n \leq 4$, and $1 < k \leq n$ and $|\mu| < 0.2$ there are some negative differences $\Delta P_a(n,k,\mu,\sigma)$, where $-0.00025 < \Delta P_a(n,k,\mu,\sigma) < 0$ (see Table 6).

Table 6

The negative $\Delta P_a(n,k,\mu,\sigma)$ and the respective p-values

	value	p-value
$\Delta P_4(2,2,0.1,1)$	-0.000138	0.214109
$\Delta P_4(3,2,0.1,1)$	-0.000247	0.035223
$\Delta P_6(3,2,0.1,1)$	-0.000176	0.053050
$\Delta P_6(4,2,0.1,1)$	-0.000076	0.258990
$\Delta P_8(2,2,0.1,1)$	-0.000106	0.148377
$\Delta P_8(2,2,0.2,1)$	-0.000066	0.278841
$\Delta P_8(4,4,0.1,1)$	-0.000140	0.056779

Furthermore (see *Supplement 2*):

1. For $a \in \{4,6,8\}$, $1 < n \leq 4$, $1 \leq k \leq n$, $0 \leq |\mu_h| < |\mu_j| \leq 6.0$ and $1.0 \leq \sigma \leq 7.0$ (see *Fig 1-31* of *Appendix)*:
   $P_{N,\alpha}(n,k,\mu_h,\sigma) < P_{N,\alpha}(n,k,\mu_j,\sigma)$
2. For $a \in \{4,6,8\}$, $1 < n \leq 4$, $1 \leq k \leq n$, $0 \leq |\mu| \leq 6.0$, $1.0 \leq \sigma_h < \sigma_j \leq 7.0$ (see *Fig 1-31* of *Appendix)*:
   $P_{N,\alpha}(n,k,\mu,\sigma_h) < P_{N,\alpha}(n,k,\mu,\sigma_j)$
3. For $a \in \{4,6,8\}$, $1 < n \leq 4$, $1 \leq k_h < k_j \leq n$, $0 \leq |\mu| \leq 6.0$, and $1.0 \leq \sigma \leq 7.0$ (see *Fig 32-39* of *Appendix)*:
   $P_{N,a}(n,k_h,\mu,\sigma) < P_{N,a}(n,k_j,\mu,\sigma)$
4. For $a_h \in \{4,6\}$, $a_h \in \{6,8\}$, $a_h < a_j$, $1 \leq n \leq 4$, $1 \leq k \leq n$, $0 < |\mu| \leq 6.0$, and $1.0 < \sigma \leq 7.0$ (see *Fig 40-49* of *Appendix)*:
   $P_{N,a_h}(n,k,\mu,\sigma) > P_{N,a_j}(n,k,\mu,\sigma)$
5. For $a \in \{4,6,8\}$, $1 \leq n_h < n_j \leq 4$, $0 < |\mu| \leq 6.0$, and $1.0 < \sigma \leq 7.0$ (see *Fig 50-52* of *Appendix)*:
   $P_{N,a}(n_h,n_h,\mu,\sigma) < P_{N,a}(n_j,n_j,\mu,\sigma)$
6. For $a \in \{4,6,8\}$, $1 \leq n_h < n_j \leq 4$, $1 \leq k \leq n_h$, $0 < |\mu| \leq 6.0$, and $1.0 < \sigma \leq 7.0$ (see *Fig 53-61* of *Appendix)*:
   $P_{N,a}(n_j,k,\mu,\sigma) < P_{N,a}(n_h,k,\mu,\sigma)$

5 Discussion

The complexity of SQC has increased significantly since the 1920s, when it was introduced into industry by Shewhart [16], especially after the development of low cost digital computers on integrated circuits with substantial processing power. Since 1989, the wide availability of more processing power led to the development of ANN and later of DNN, including CNN, and their application to SQC. Deep learning transforms low level representations to a higher order abstract one, extracting features from raw data in multiple stages. CNN are relatively simple and compact DNN, with exemplary learning ability and performance [4]. Particularly, 1-D CNN simple enough to be implemented in low cost computer systems have been successfully used to process 1-D signals [5, 17], including control charts [18].

Although the relationship among a QC procedure, the reliability of a system, the risk of nonconformity to quality specifications and the QC related cost is complex [19], it is obvious that cost increases with the number of the QC measurements required for the safe detection of nonconformity. The application of statistical QC functions to very small samples of QC measurements has been extensively studied [20] [21]. However, ANN QC functions have only been applied to n-tuples of QC measurements with $n > 5$. Consequently, it is meaningful to explore their application to n-tuples of QC measurements with $1 \leq n \leq 4$.

For this project, there were designed 4 1-D CNN with 1x1 kernels, applicable to 1 to 4 QC measurements respectively. As the directly cost related probability for false rejection is a key index of the performance of a QC function, each of the 4 1-D CNN was trained with three training datasets, with different ratios of n-tuples of random simulated QC measurements distributed as $\mathcal{N}(0,1)$ (see *Supplement 1*). As a result, 12 trained 1-D CNN were obtained, which were applied to testing datasets containing all the possible combinations of measurements. Their probabilities for false rejection were from 0.005599 to 0.023750 (see Table 3). Another approach for obtaining varying probabilities for false rejection could be weighing the output of the CNN, with differing penalization of the probability for false rejection

The comparison of the trained 1-D CNN with the respective statistical QC functions with decision limits shown on Table 3, shows that the designed CNN outperform the respective statistical QC functions, for $2 \leq n \leq 4$ and $2 \leq k \leq n$, that is when the samples they are applied to include at least two QC measurements out of control, distributed as $\mathcal{N}(\mu, \sigma^2)$, for $0.2 < |\mu| \leq 6.0$, and $1.0 < \sigma \leq 7.0$. However, statistical QC functions outperform the 1-D CNN, for $1 \leq n \leq 4$ and $k = 1$, that is when the samples they are applied to include only one control measurement distributed as $\mathcal{N}(\mu, \sigma^2)$, but then $|\Delta P_a(n, 1, \mu, \sigma)| < 0.03$ (see *Online Resources 3* and *4*).

Besides, the probability for rejection of the 1-D CNN increases with the sample size, when all the QC measurements of the sample are out of control, while decreases with the sample size when the numbers of the out of control measurements of each sample are equal.

Limitations of this project, that could be improved by further research, are the following:

1. The assumption of normality of either the measurements or their applicable transforms [22–25], however, this is usually valid. Furthermore, the four designed CNN can be trained and tested using datasets of any parametric or nonparametric distribution.
2. The arbitrary selection of the different ratios of n-tuples of random simulated QC measurements distributed as $\mathcal{N}(0,1)$ of the simulated datasets, for obtaining varying probabilities for false rejection. Another approach could be weighing the output of the CNN with differing penalization of the probability for false rejection.
3. The arbitrary design of the four CNN. As the processing power of the computer systems is increasing exponentially, efficient methods of optimal design of ANN will be developed, including evolutionary algorithms [26].

The twelve trained CNN can be used as they are or they can be trained with different datasets. Their models, including their weights and biases, are available in various formats, to be implemented in any neural networks' framework (see in *Supplement 1*).

6 Conclusion

One-dimensional convolutional neural networks applied to samples of 2-4 QC measurements can be used in quality control, to improve the detection of nonconformity of a process to quality specifications, with lower cost.

7 References

1. Hatjimihail AT (1992) A tool for the design and evaluation of alternative quality-control procedures. Clin Chem 38:204–210. https://doi.org/10.1093/clinchem/38.2.204

2. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115–133. https://doi.org/10.1007/BF02478259

3. Leijnen S, van Veen F (2020) The Neural Network Zoo. Proc AMIA Annu Fall Symp 47:9. https://doi.org/10.3390/proceedings2020047009

4. Khan A, Sohail A, Zahoora U, Qureshi AS (2020) A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review 53:5455–5516. https://doi.org/10.1007/s10462-020-09825-6

5. Kiranyaz S, Avci O, Abdeljaber O, et al (2021) 1D convolutional neural networks and applications: A survey. Mech Syst Signal Process 151:107398. https://doi.org/10.1016/j.ymssp.2020.107398

6. Zorriassatine F, Tannock JDT (1998) A review of neural networks for statistical process control. J Intell Manuf 9:209–224. https://doi.org/10.1023/A:1008818817588

7. Hatjimihail AT (1993) Genetic algorithms-based design and optimization of statistical quality-control procedures. Clin Chem 39:1972–1978

8. Pugh GA (1989) Synthetic neural networks for process control. Comput Ind Eng 17:24–26. https://doi.org/10.1016/0360-8352(89)90030-2

9. Hwarng HB, Hubele NF (1993) X Control Chart Pattern Identification through Efficient Off-Line Neural Network Training. IIE Trans 25:27–40. https://doi.org/10.1080/07408179308964288

10. Smith AE (1994) X-bar and R control chart interpretation using neural computing. Int J Prod Res 32:309–320. https://doi.org/10.1080/00207549408956935

11. Stützle T (1995) A neural network approach to Quality Control Charts. In: From Natural to Artificial Neural Computation. Springer Berlin Heidelberg, pp 1135–1141

12. Cheng C-S (1995) A multi-layer neural network model for detecting changes in the process mean. Comput Ind Eng 28:51–61. https://doi.org/10.1016/0360-8352(94)00024-H

13. Chang SI, Aw CA (1996) A neural fuzzy control chart for detecting and classifying process mean shifts. Int J Prod Res 34:2265–2278. https://doi.org/10.1080/00207549608905024

14. Psarakis S (2011) The use of neural networks in statistical process control charts. Qual Reliab Eng Int 27:641–650. https://doi.org/10.1002/qre.1227

15. Hachicha W, Ghorbel A (2012) A survey of control-chart pattern-recognition literature (1991–2010) based on a new conceptual classification scheme. Comput Ind Eng 63:204–222. https://doi.org/10.1016/j.cie.2012.03.002

16. Shewhart WA (1930) Economic quality control of manufactured Product. D. Van Nostrand Company, Inc., New York

17. Kiranyaz S, Ince T, Gabbouj M (2016) Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks. IEEE Transactions on Biomedical Engineering 63:664–675. https://doi.org/10.1109/TBME.2015.2468589

18. Xu J, Lv H, Zhuang Z, et al (2019) Control Chart Pattern Recognition Method Based on Improved One-dimensional Convolutional Neural Network. IFAC-PapersOnLine 52:1537–1542. https://doi.org/10.1016/j.ifacol.2019.11.418

19. Hatjimihail AT (2009) Estimation of the optimal statistical quality control sampling time intervals using a residual risk measure. PLoS One 4:e5770. https://doi.org/10.1371/journal.pone.0005770

20. Duncan AJ (1986) Quality control and industrial statistics. 5th edition. Irwin, Homewood, Illinois

21. Montgomery DC (2019) Introduction to Statistical Quality Control, 8th ed. Wiley

22. Gillard J (2012) A generalised Box–Cox transformation for the parametric estimation of clinical reference intervals. J Appl Stat 39:2231–2245. https://doi.org/10.1080/02664763.2012.706266

23. Atkinson AC, Riani M, Corbellini A (2021) The Box–Cox Transformation: Review and Extensions. SSO Schweiz Monatsschr Zahnheilkd 36:239–255. https://doi.org/10.1214/20-STS778

24. Sakia RM (1992) The Box-Cox Transformation Technique: A Review. Journal of the Royal Statistical Society Series D (The Statistician) 41:169–178. https://doi.org/10.2307/2348250

25. Box GEP, Cox DR (1964) An Analysis of Transformations. J R Stat Soc Series B Stat Methodol 26:211–243. https://doi.org/10.1111/j.2517-6161.1964.tb00553.x

26. Ding S, Li H, Su C, et al (2013) Evolutionary artificial neural networks: a review. Artificial Intelligence Review 39:251–260. https://doi.org/10.1007/s10462-011-9270-6

8. Formalisms and Notation

8.1 Abbreviations

QC: quality control

ANN: artificial neural network(s)

DNN: deep neural network(s)

CNN: convolutional neural network(s)

1-D CNN: one dimensional convolutional neural network(s)

GA: genetic algorithms

QCMT: quality control measurements tuple(s)

8.2 Operators

 # : cardinality

$\sim$: distributed as

8.3 Datasets

$\mathbf{Z}_\Theta = \{\boldsymbol{z}_i\}_{i=1}^{n} = \{\boldsymbol{x}_i, \mathrm{y}_i\}_{i=1}^{n} = \{\boldsymbol{x}_i, f(\boldsymbol{x}_i;\Theta)\}_{i=1}^{n}$*: dataset of* n*-tuples* $\boldsymbol{x_i}$ *of QC measurements with* $y_\mathrm{i} \in \{True, False\}$

$\mathbf{T}_a(n)$*: training dataset of* n*-tuples* $\boldsymbol{x}_i$ *of QC measurements.*

$\mathbf{D}(n, 0{,}1)$*: testing dataset of* n*-tuples* $\boldsymbol{x}_i$ *of QC measurements, distributed as* $\mathcal{N}(0{,}1)$*.*

$\mathbf{D}(n, k, \mu, \sigma)$*: testing dataset of* n*-tuples* $\boldsymbol{x}_i$ *of QC measurements, where* k *of them are distributed as* $\mathcal{N}(\mu, \sigma^2)$ *and (*n *-* k*) as* $\mathcal{N}$ *(0,1).*

8.4 Functions

$\binom{n}{k}$: binomial coefficient

$\mathcal{N}(\mu, \sigma^2)$: normal distribution with mean μ and variance σ^2

$\mathrm{N_a}(\boldsymbol{x})$*: neural network QC function applied to a tuple* $\boldsymbol{x}$ *of QC measurements;* $\mathrm{N_a}(\boldsymbol{x}) \in \{True, False\}$

$\mathrm{S}_\alpha(\boldsymbol{x};\ \theta) = \mathrm{S_a}(\boldsymbol{x}; l, \mathrm{m}, \mathrm{s})$*: statistical QC function, applied to a tuple* $\boldsymbol{x}$ *of QC measurements;*

$$\mathrm{S_a}(\boldsymbol{x}; l, \mathrm{m}, \mathrm{s}) = \begin{cases} True \text{ if } \exists\, \mathrm{x_i} \in \boldsymbol{x}, |x_i - \mathrm{m}| \geq l\,\mathrm{s} \\ False \text{ otherwise} \end{cases}$$

$L(f_q; \theta) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{I}\big(f_q(\boldsymbol{x}_i; \theta) \neq f(\boldsymbol{x}_i; \theta)\big)$: misclassification rate of QC function f_q, where $I(e) = \begin{cases} 1 \text{ if } e\ True, \\ 0 \text{ if } e\ False \end{cases}$'

$$f(\boldsymbol{x}; \theta) = \begin{cases} True \text{ if } \exists\, \mathrm{x_i} \in \boldsymbol{x}, X_i \sim \mathcal{N}(\mu, \sigma^2) \wedge (\mu > 0 \vee \sigma > 1), \\ \qquad False \text{ if } \forall\, x_i \in \boldsymbol{x}, X_i \sim \mathcal{N}(0{,}1) \end{cases} \quad \textit{and } f_q(\boldsymbol{x};\ \theta) = \begin{cases} \mathrm{N}_\alpha(\boldsymbol{x}) \\ \vee \\ \mathrm{S}_\alpha(\boldsymbol{x};\ \theta) \end{cases}$$

$P_{N,a}(n, 0{,}1)$: probability for false rejection of the $N_\mathrm{a}(\boldsymbol{x})$ *applied* to the dataset $\boldsymbol{D}(n, 0{,}1)$ for $n = \#\boldsymbol{x}$*, and* $1 \leq n \leq 4$*.*
$P_{N,a}(n, k, \mu, \sigma)$: probability for rejection of the $N_\mathrm{a}(\boldsymbol{x})$ *applied* to the dataset $\boldsymbol{D}(n, k, \mu, \sigma)$ for $n = \#\boldsymbol{x}$, $1 \leq n \leq 4$, and $1 \leq k \leq n$.
$P_{S,a}(n, 0{,}1)$: probability for rejection of the $S_\mathrm{a}(\boldsymbol{x}; l, 0{,}1)$ applied to the dataset $\boldsymbol{D}(n, 0{,}1)$ for $n = \#\boldsymbol{x}$*,* $1 \leq n \leq 4$, and decision limit l*.*
$P_{S,a}(n, k, \mu, \sigma)$: probability for rejection of the $S_\mathrm{a}(\boldsymbol{x}; l, 0{,}1)$ applied to the dataset $\boldsymbol{D}(n, k, \mu, \sigma)$ for $n = \#\ \boldsymbol{x}$, $1 \leq n \leq 4$, $1 \leq k \leq n$, and decision limit l*.*
The decision limit l is defined as following.
Let $P_{C,a}(\boldsymbol{x}; k, l, \mu, \sigma)$ be the probability for rejection of a n*-tuple of QC measurements, for* $n = \#\boldsymbol{x}$*,* $1 \leq n \leq 4$, $1 \leq k \leq n$*, where* k *of them are distributed as* $\mathcal{N}(\mu, \sigma^2)$ *and (*n *-* k*) as* $\mathcal{N}$*(0,1). Then:*

$$P_{C,a}(\boldsymbol{x}; k, l, \mu, \sigma) = 1 - \frac{1}{2^k}\mathrm{erf}\Big(\frac{1}{\sqrt{2}}\Big)^{n-k}\left(\mathrm{erf}\Big(\frac{\mu+l}{\sigma\sqrt{2}}\Big) - \mathrm{erf}\Big(\frac{\mu-l}{\sigma\sqrt{2}}\Big)\right)^{k} \text{ for } 0 > k \leq n$$

To define l, we solve the following equation:

$$P_{C,a}(\boldsymbol{x}; n, l, 0,1) = P_{N,a}(n, 0,1) \Rightarrow l = \sqrt{2}\,\mathrm{erf}^{-1}\left(\left(1 - P_N(n, 0,1)\right)^{\frac{1}{n}}\right)$$

Therefore:

$$P_{S,a}(n, k, \mu, \sigma) = P_{C,a}\left(\boldsymbol{x}; k, \sqrt{2}\,\mathrm{erf}^{-1}\left(\left(1 - P_{N,a}(n, 0,1)\right)^{\frac{1}{n}}\right), \mu, \sigma\right)$$

In addition:

$$\Delta P_a(n, \mu, \sigma) = P_{N,a}(n, \mu, \sigma) - P_{S,a}(n, \mu, \sigma)$$

$$\Delta P_a(n, k, \mu, \sigma) = P_{N,a}(n, k, \mu, \sigma) - P_{S,a}(n, k, \mu, \sigma)$$

$$\Delta P_{R,a}(n, k, \mu, \sigma) = \frac{\Delta P_a(n, k, \mu, \sigma)}{P_{S,a}(n, k, \mu, \sigma)}$$

9. Supplementary Information

The following resources are available online:

1. *Supplement 1,* with the 1D - CNN models in various formats, at: https://www.hcsl.com/Supplements/hcsltr21s1.zip
2. *Supplement 2,* with spreadsheets presenting the probabilities for rejection $P_{N,a}(n, k, \mu, \sigma)$ and $P_{S,a}(n, k, \mu, \sigma)$, their differences $\Delta \mathrm{P}_a(n, k, \mu, \sigma)$ and the respective statistical significance (p-values), and their relative differences $\Delta \mathrm{P}_{R,a}(n, k, \mu, \sigma)$, at: https://www.hcsl.com/Supplements/hcsltr21s2.zip

Appendix

Plots of the Probabilities for Rejection of Quality Control Functions

Table of Figures

Fig 21: $P_{N,8}(1,1,\mu,\sigma)$ vs $P_{S,8}(1,1,\mu,\sigma)$ 38
Fig 22: $P_{N,8}(2,1,\mu,\sigma)$ vs $P_{S,8}(2,1,\mu,\sigma)$ 39
Fig 23: $P_{N,8}(2,2,\mu,\sigma)$ vs $P_{S,8}(2,2,\mu,\sigma)$ 40
Fig 24: $P_{N,8}(3,1,\mu,\sigma)$ vs $P_{S,8}(3,1,\mu,\sigma)$ 41
Fig 25: $P_{N,8}(3,2,\mu,\sigma)$ vs $P_{S,8}(3,2,\mu,\sigma)$ 42
Fig 26: $P_{N,8}(3,3,\mu,\sigma)$ vs $P_{S,8}(3,3,\mu,\sigma)$ 43
Fig 27: $P_{N,8}(4,1,\mu,\sigma)$ vs $P_{S,8}(4,1,\mu,\sigma)$ 44
Fig 28: $P_{N,8}(4,2,\mu,\sigma)$ vs $P_{S,8}(4,2,\mu,\sigma)$ 45
Fig 29: $P_{N,8}(4,3,\mu,\sigma)$ vs $P_{S,8}(4,3,\mu,\sigma)$ 46
Fig 30: $P_{N,8}(4,4,\mu,\sigma)$ vs $P_{S,8}(4,4,\mu,\sigma)$ 47

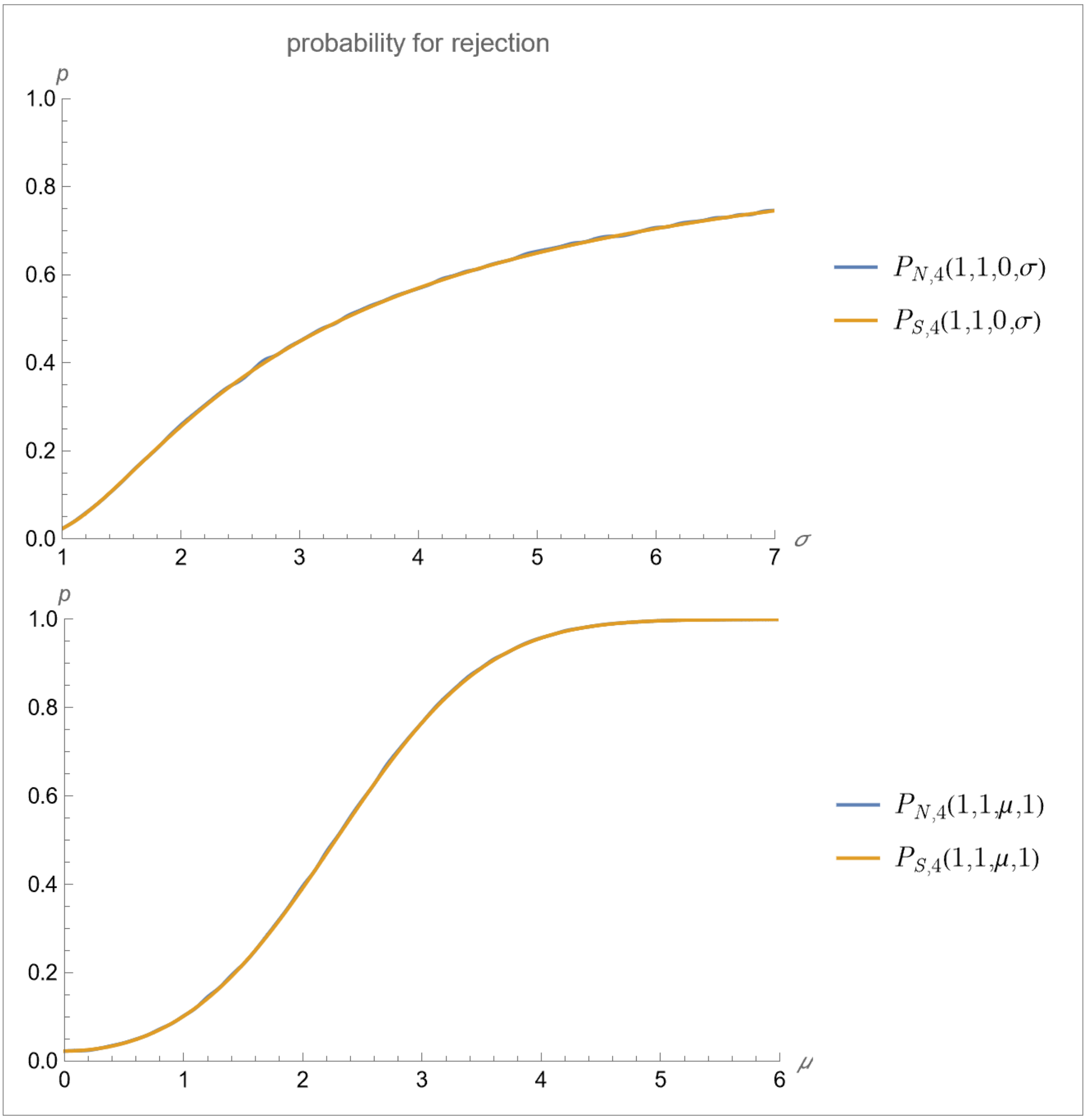

Fig 1: $P_{N,4}(1,1,\mu,\sigma)$ vs $P_{S,4}(1,1,\mu,\sigma)$

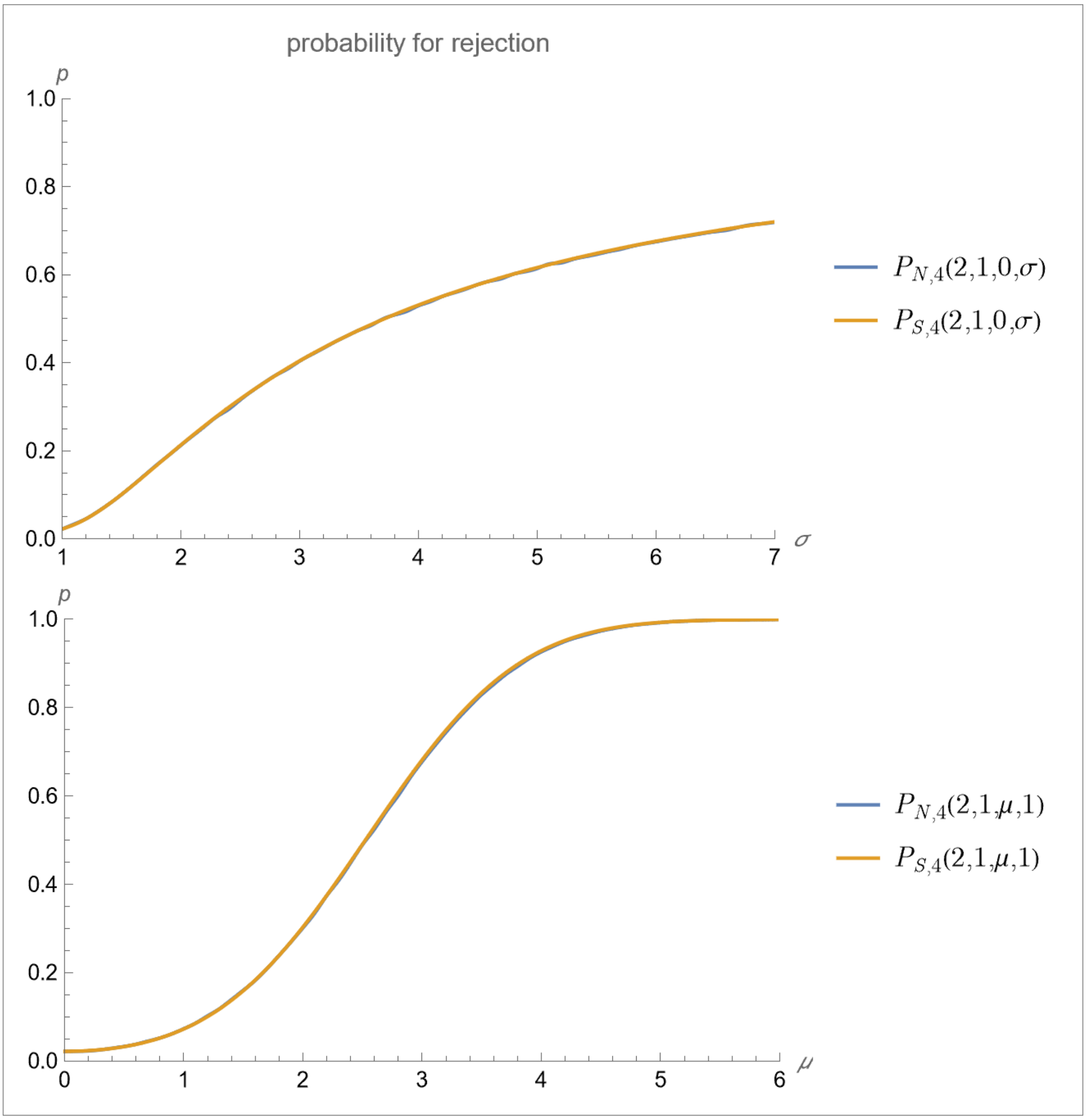

Fig 2: $P_{N,4}(2,1,\mu,\sigma)$ *vs* $P_{S,4}(2,1,\mu,\sigma)$

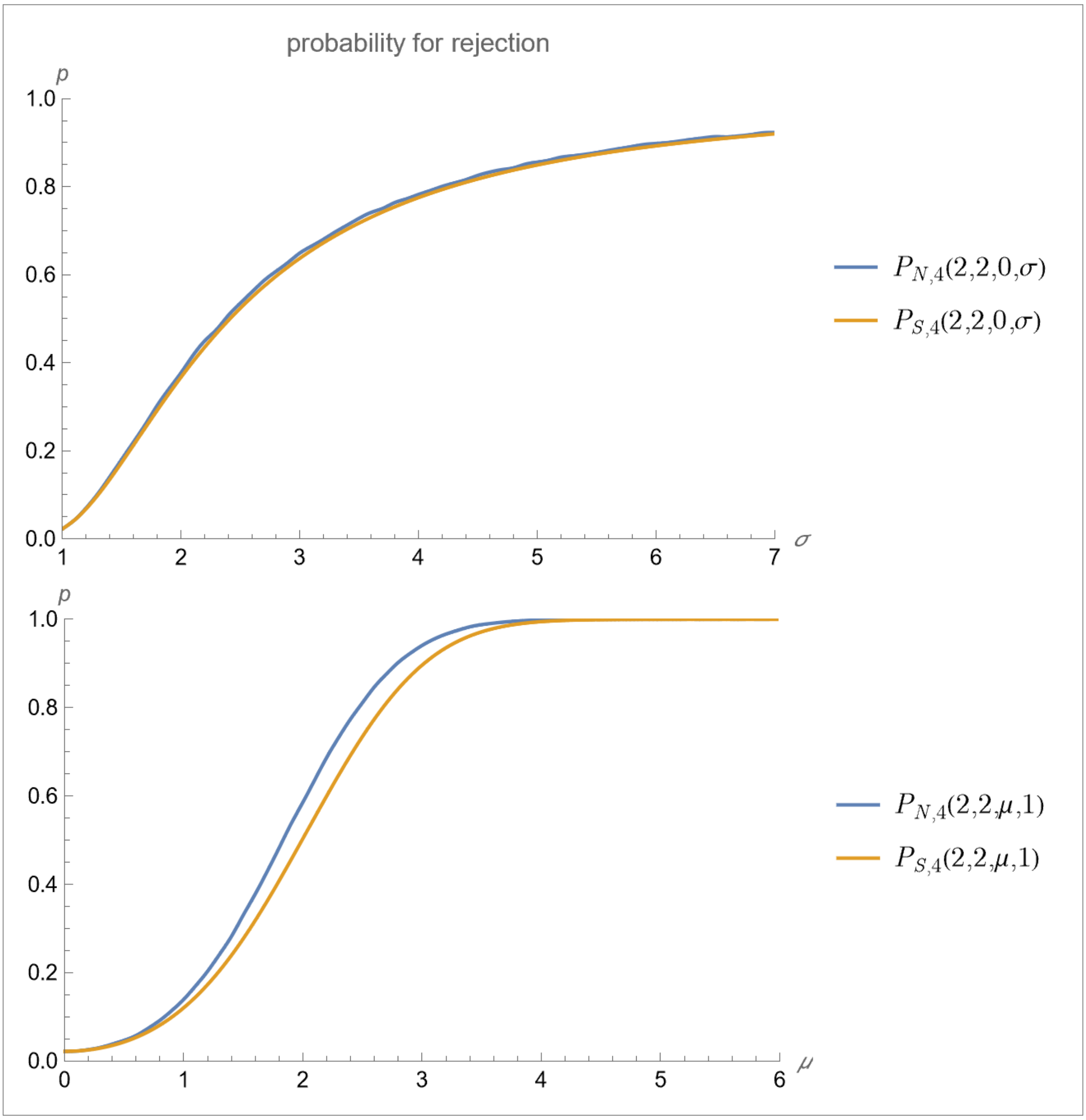

Fig 3: $P_{N,4}(2,2,\mu,\sigma)$ *vs* $P_{S,4}(2,2,\mu,\sigma)$

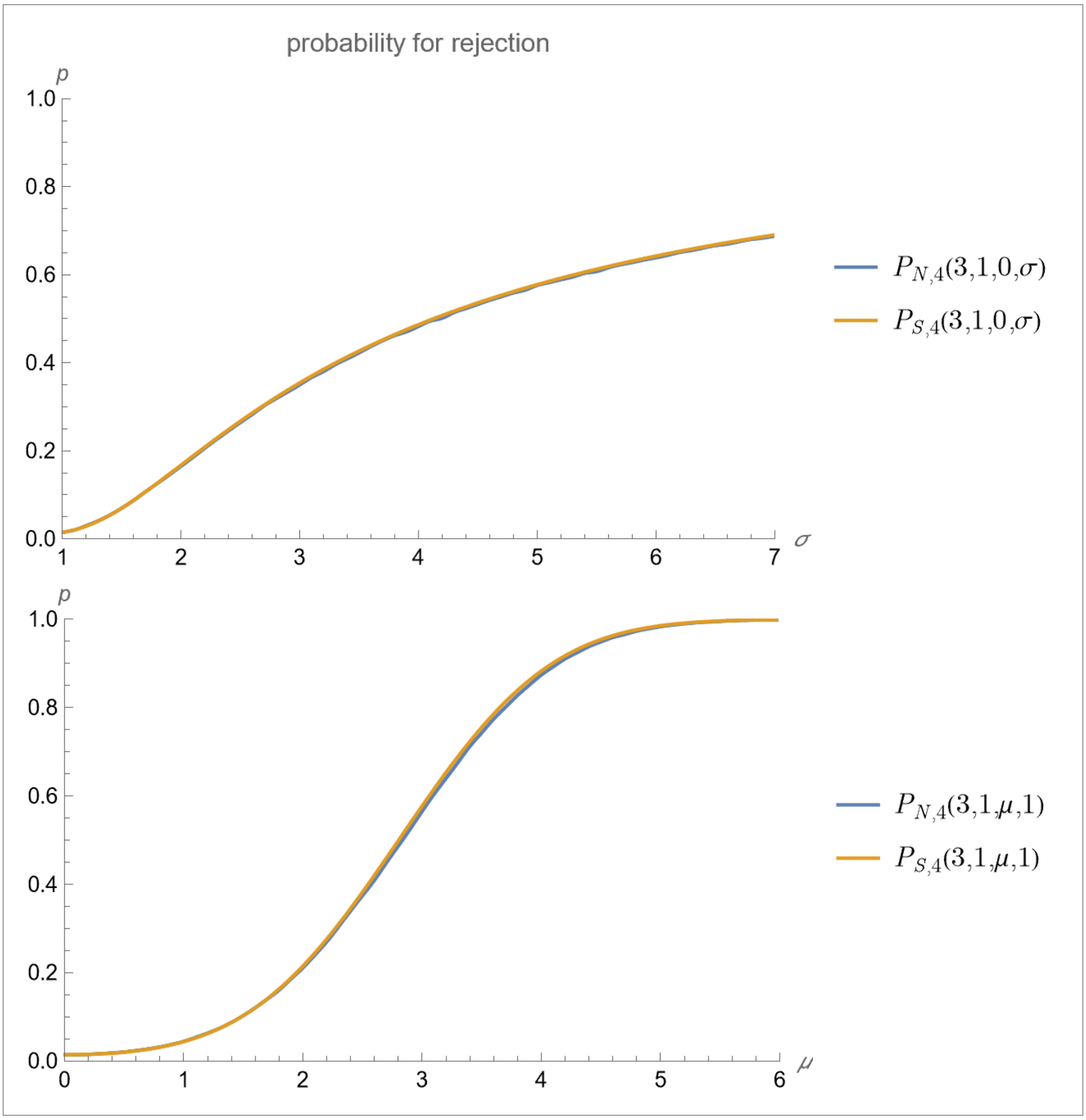

Fig 4: $P_{N,4}(3,1,\mu,\sigma)$ *vs* $P_{S,4}(3,1,\mu,\sigma)$

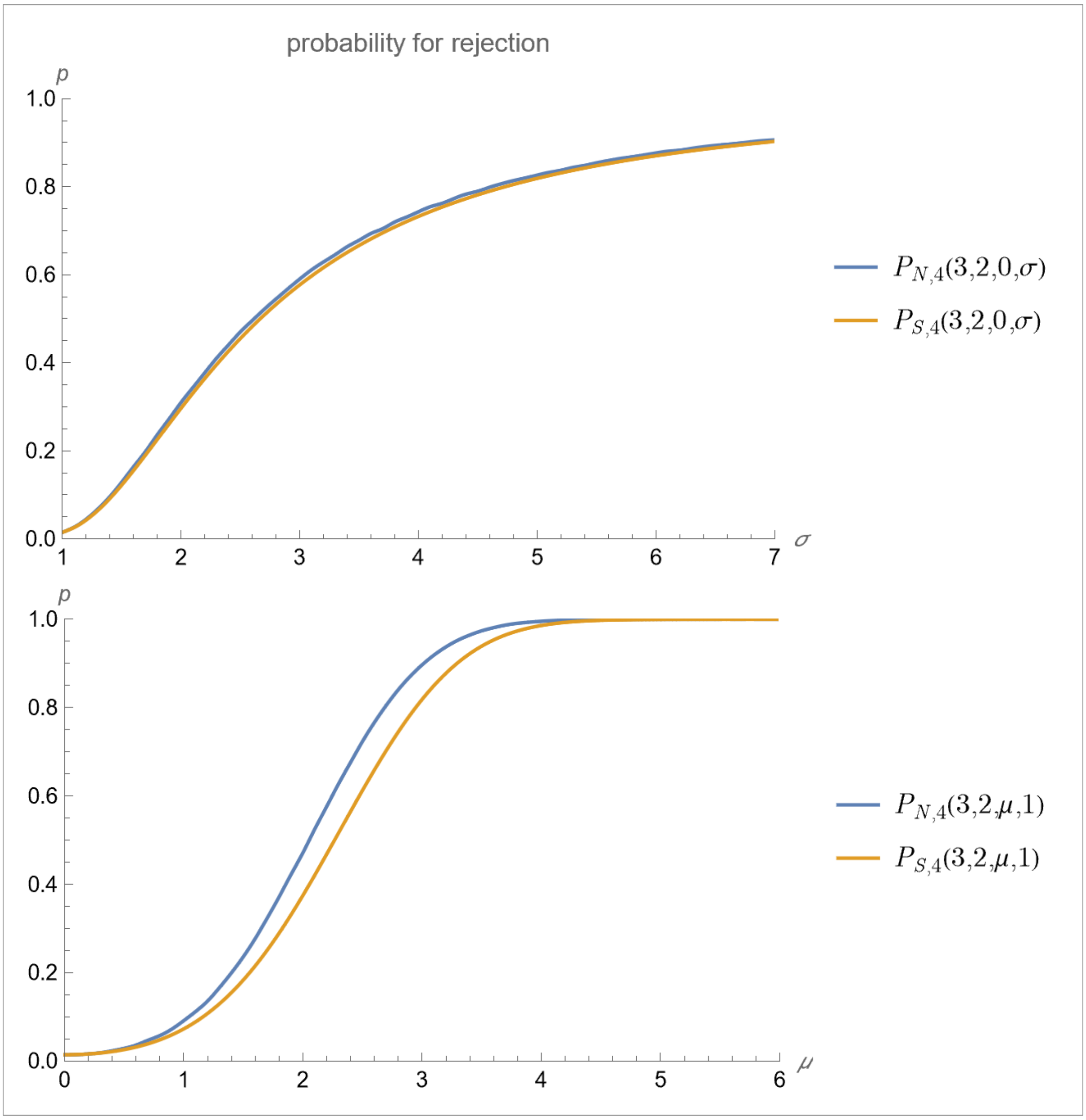

Fig 5: $P_{N,4}(3,2,\mu,\sigma)$ *vs* $P_{S,4}(3,2,\mu,\sigma)$

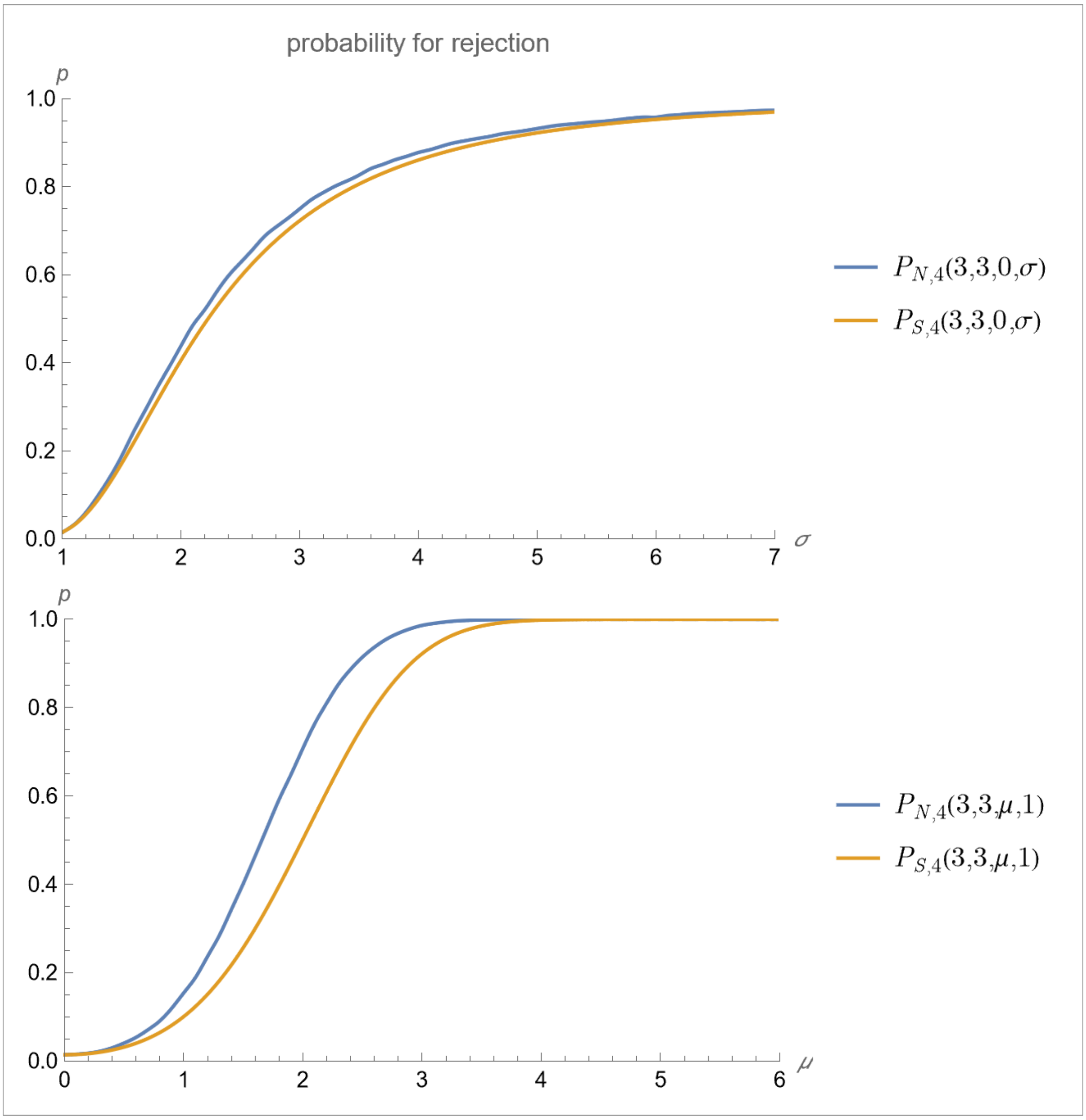

Fig 6: $P_{N,4}(3,3,\mu,\sigma)$ *vs* $P_{S,4}(3,3,\mu,\sigma)$

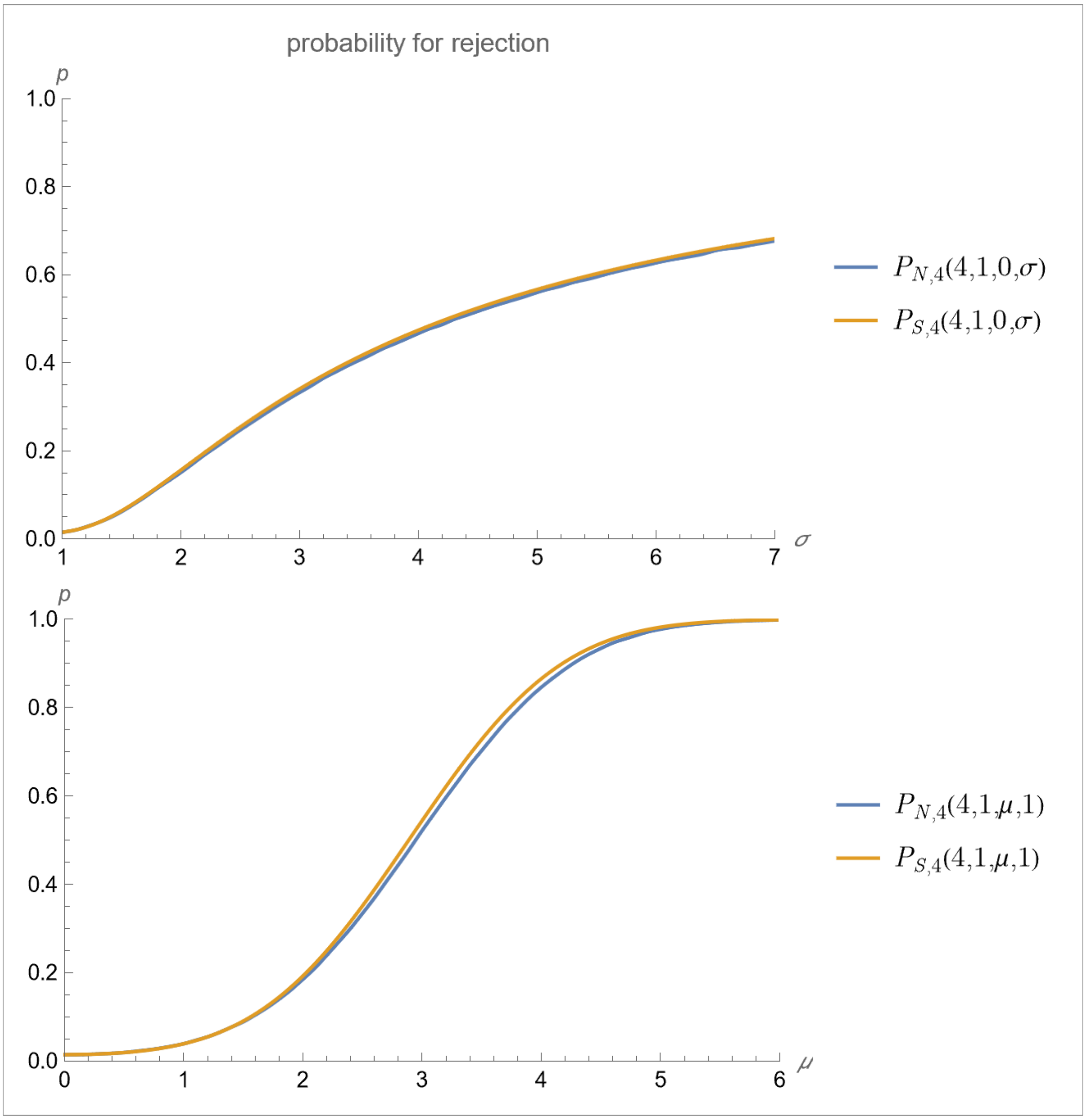

Fig 7: $P_{N,4}(4,1,\mu,\sigma)$ *vs* $P_{S,4}(4,1,\mu,\sigma)$

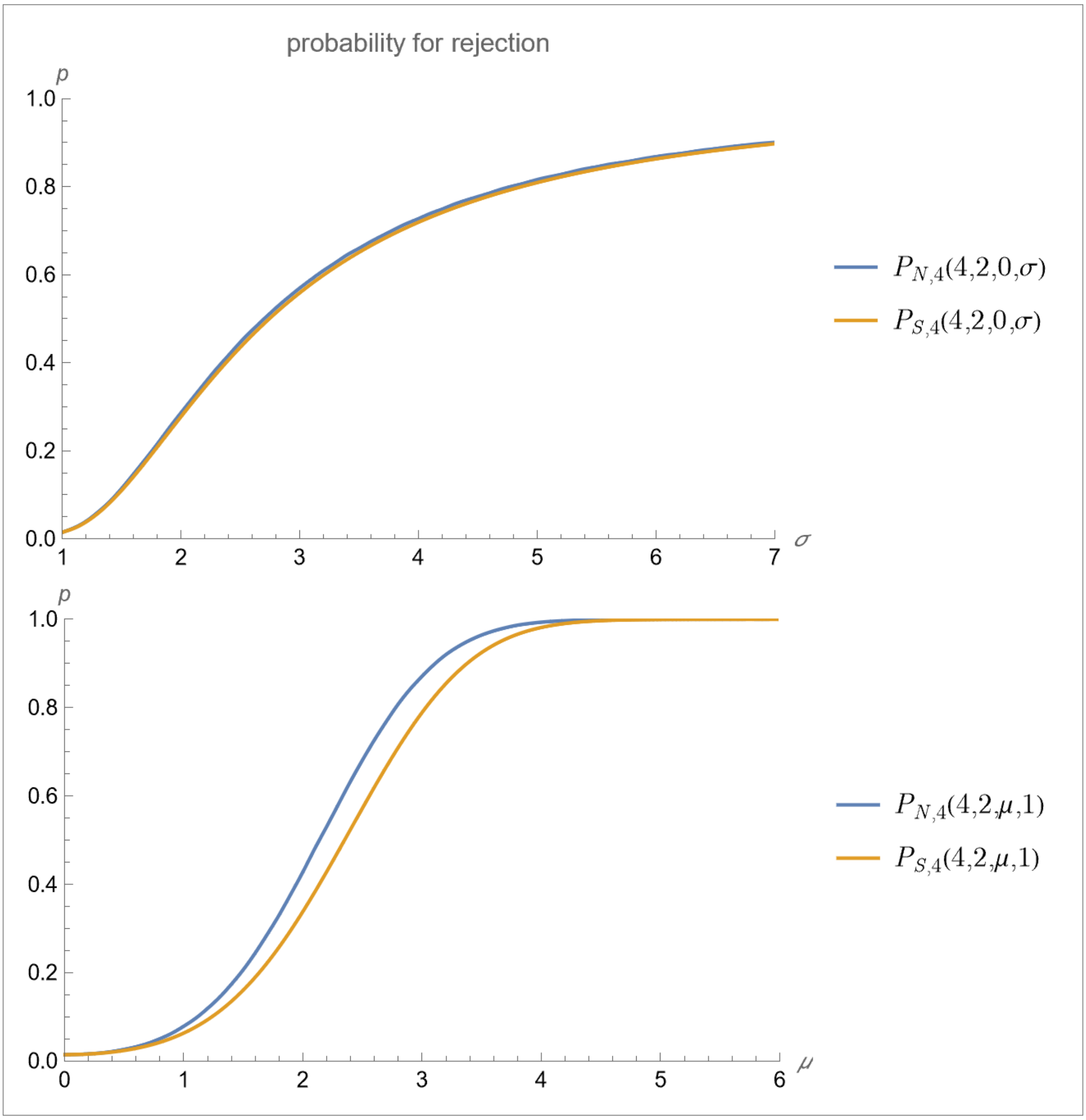

Fig 8: $P_{N,4}(4,2,\mu,\sigma)$ *vs* $P_{S,4}(4,2,\mu,\sigma)$

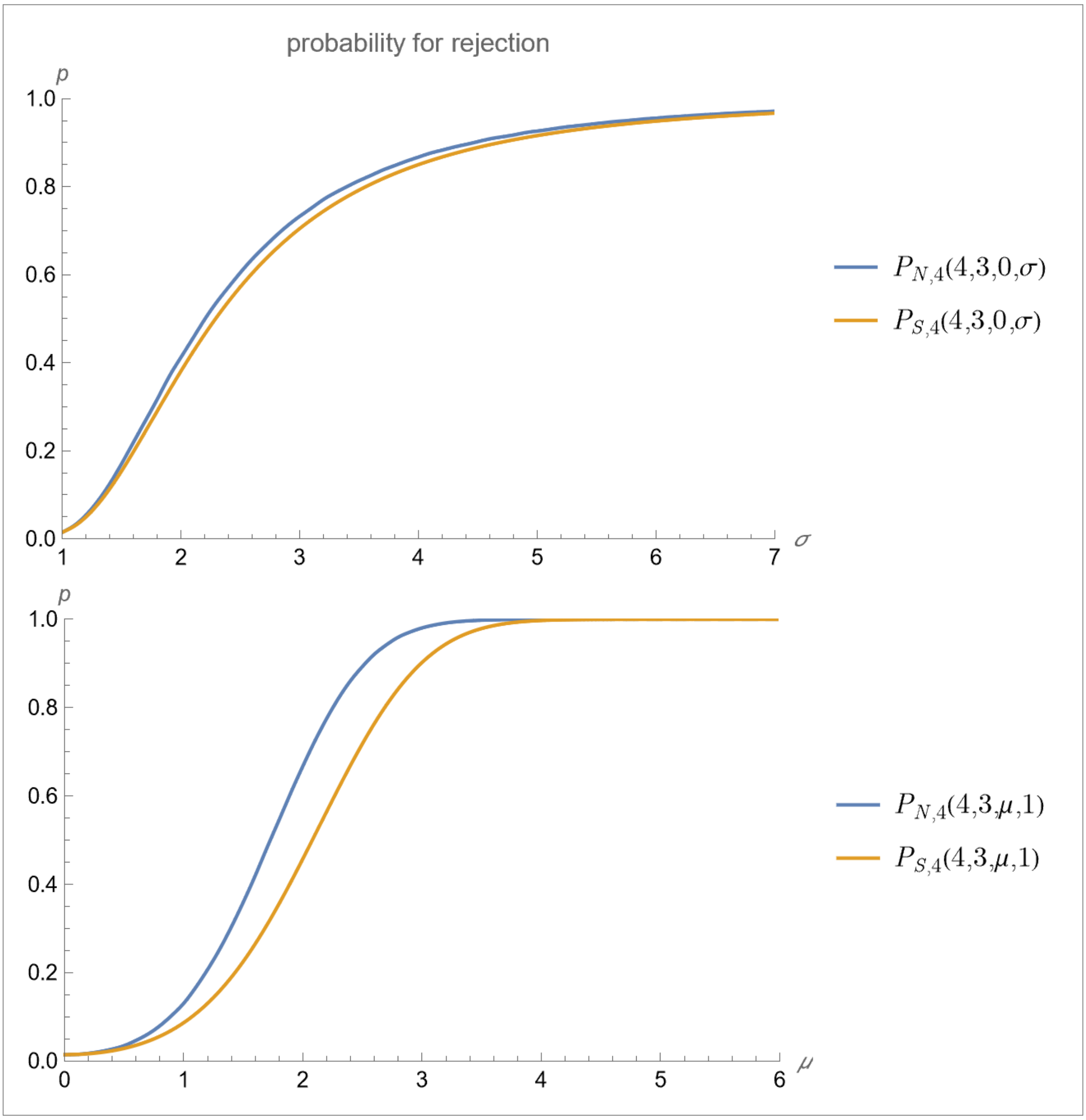

Fig 9: $P_{N,4}(4{,}3, \mu, \sigma)$ *vs* $P_{S,4}(4{,}3, \mu, \sigma)$

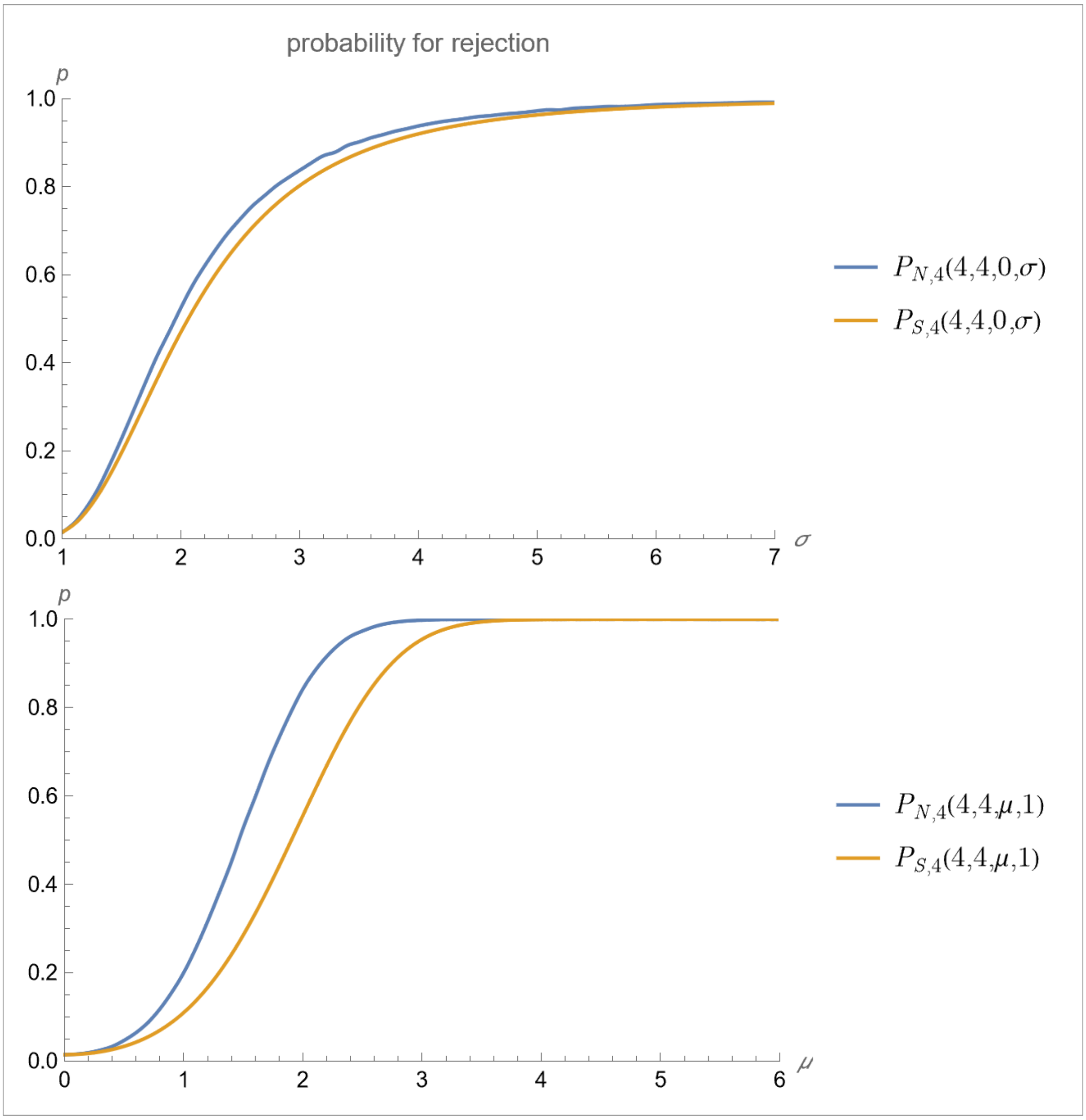

Fig 10: $P_{N,4}(4,4,\mu,\sigma)$ *vs* $P_{S,4}(4,4,\mu,\sigma)$

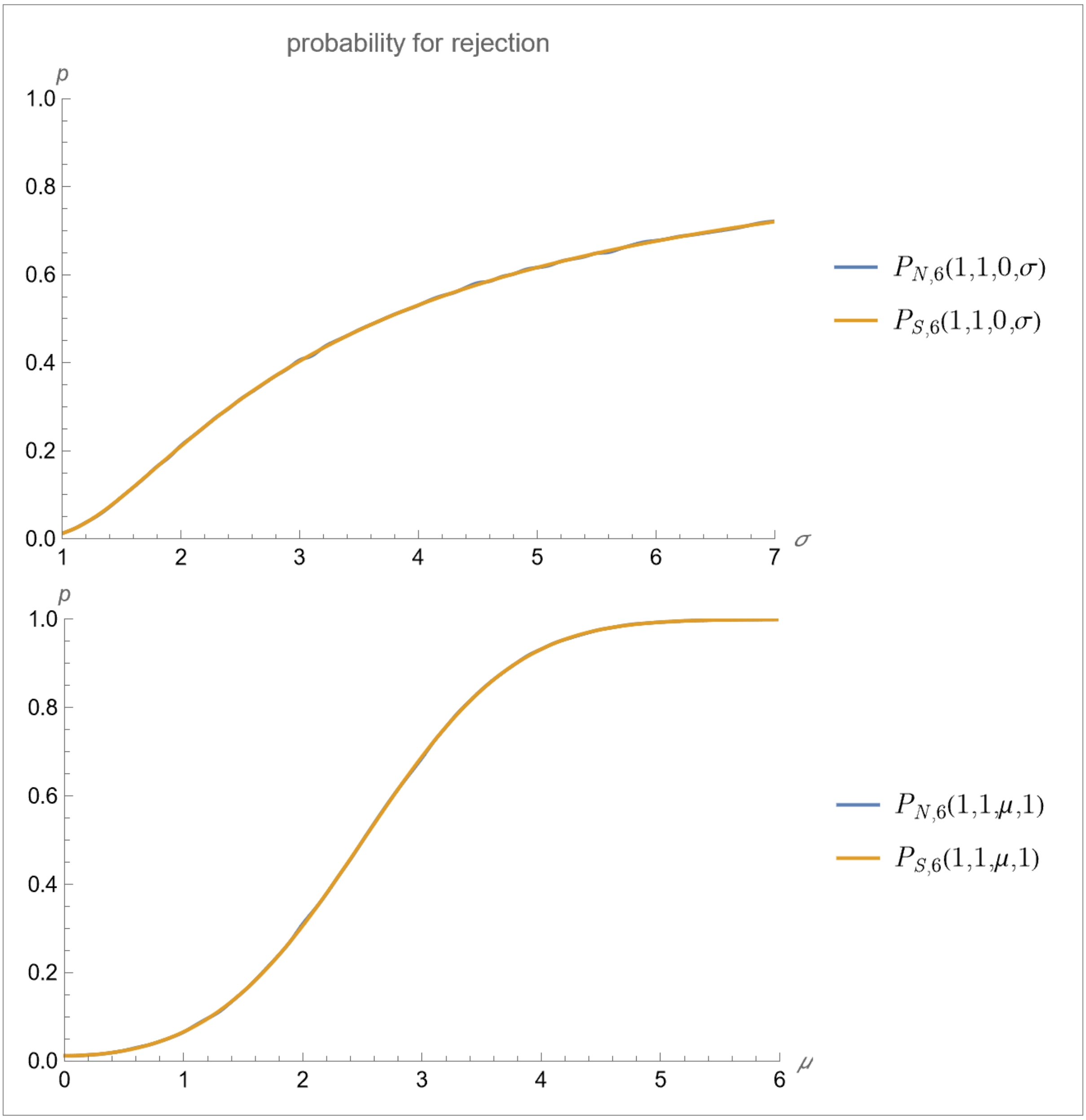

Fig 11: $P_{N,6}(1,1,\mu,\sigma)$ *vs* $P_{S,6}(1,1,\mu,\sigma)$

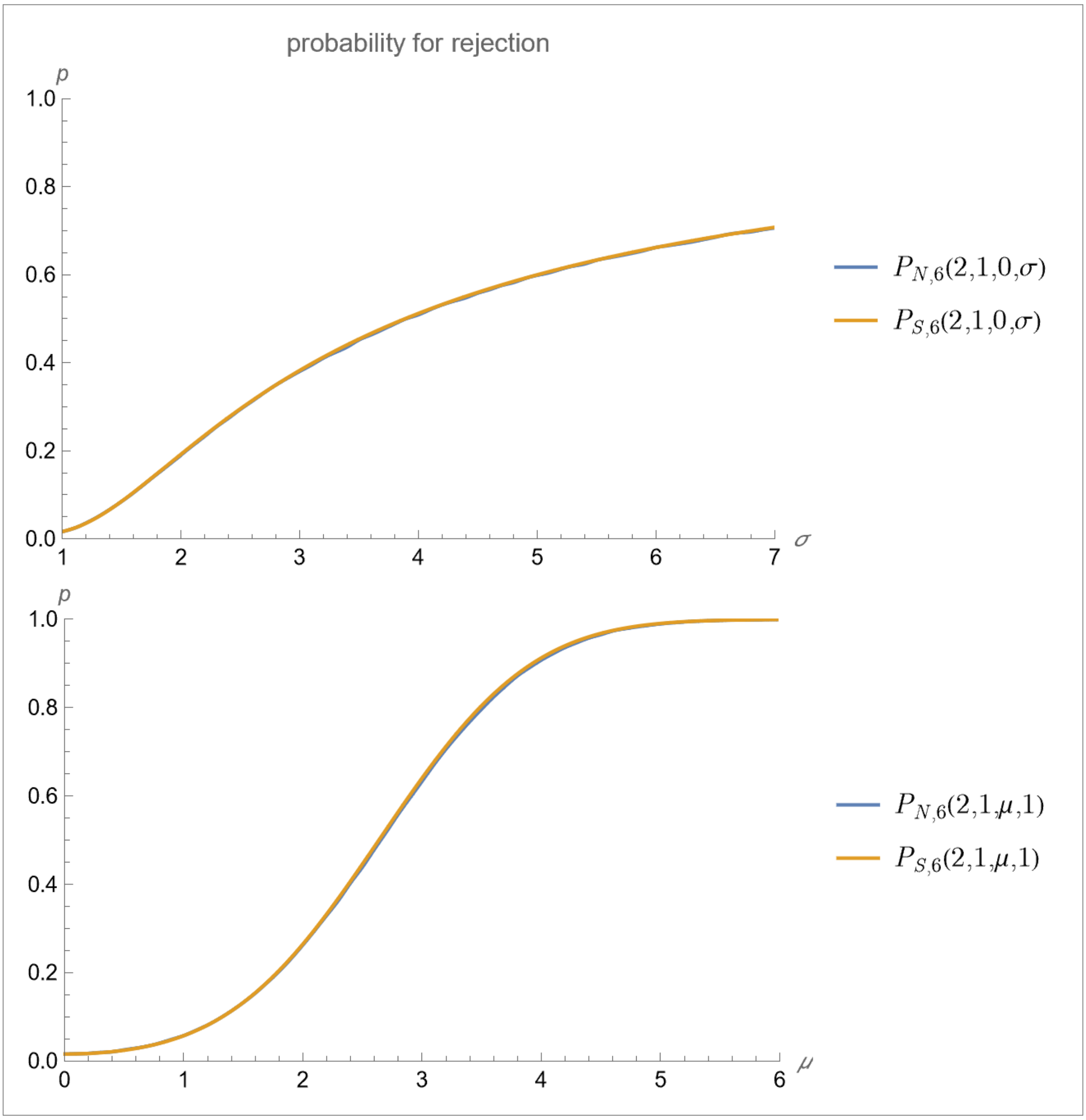

Fig 12: $P_{N,6}(2,1,\mu,\sigma)$ *vs* $P_{S,6}(2,1,\mu,\sigma)$

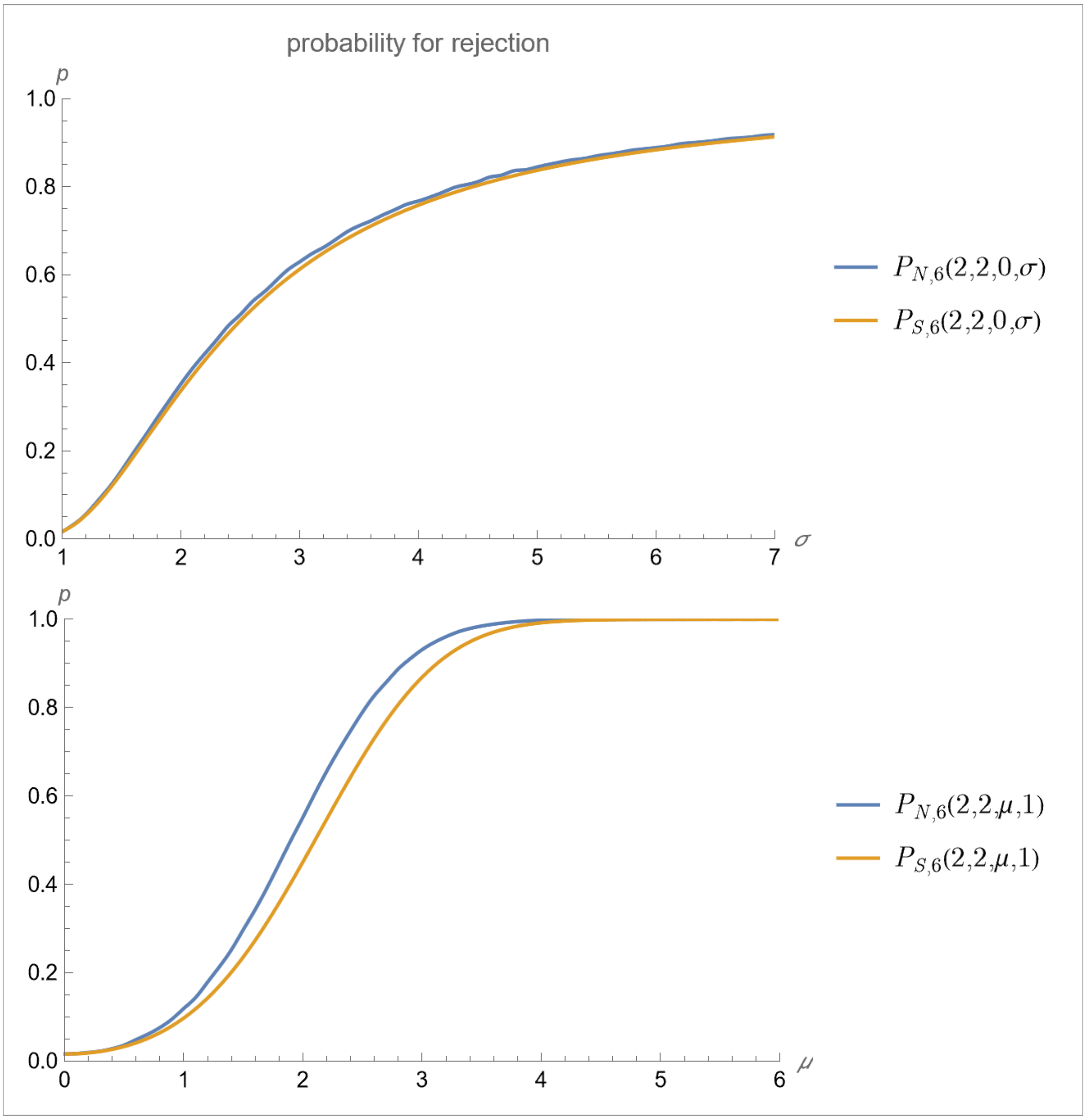

Fig 13: $P_{N,6}(2,2,\mu,\sigma)$ *vs* $P_{S,6}(2,2,\mu,\sigma)$

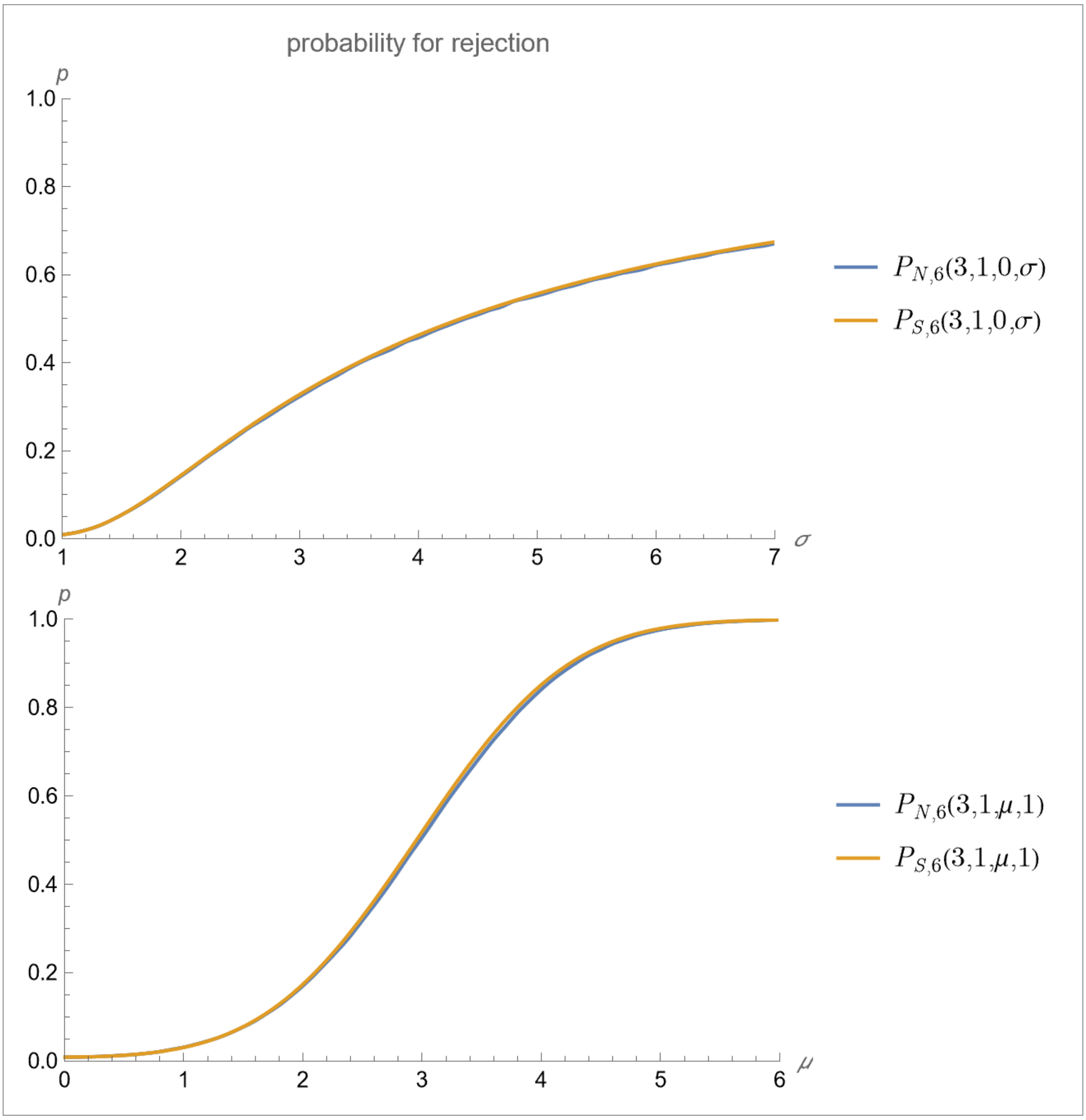

Fig 14: $P_{N,6}(3,1,\mu,\sigma)$ *vs* $P_{S,6}(3,1,\mu,\sigma)$

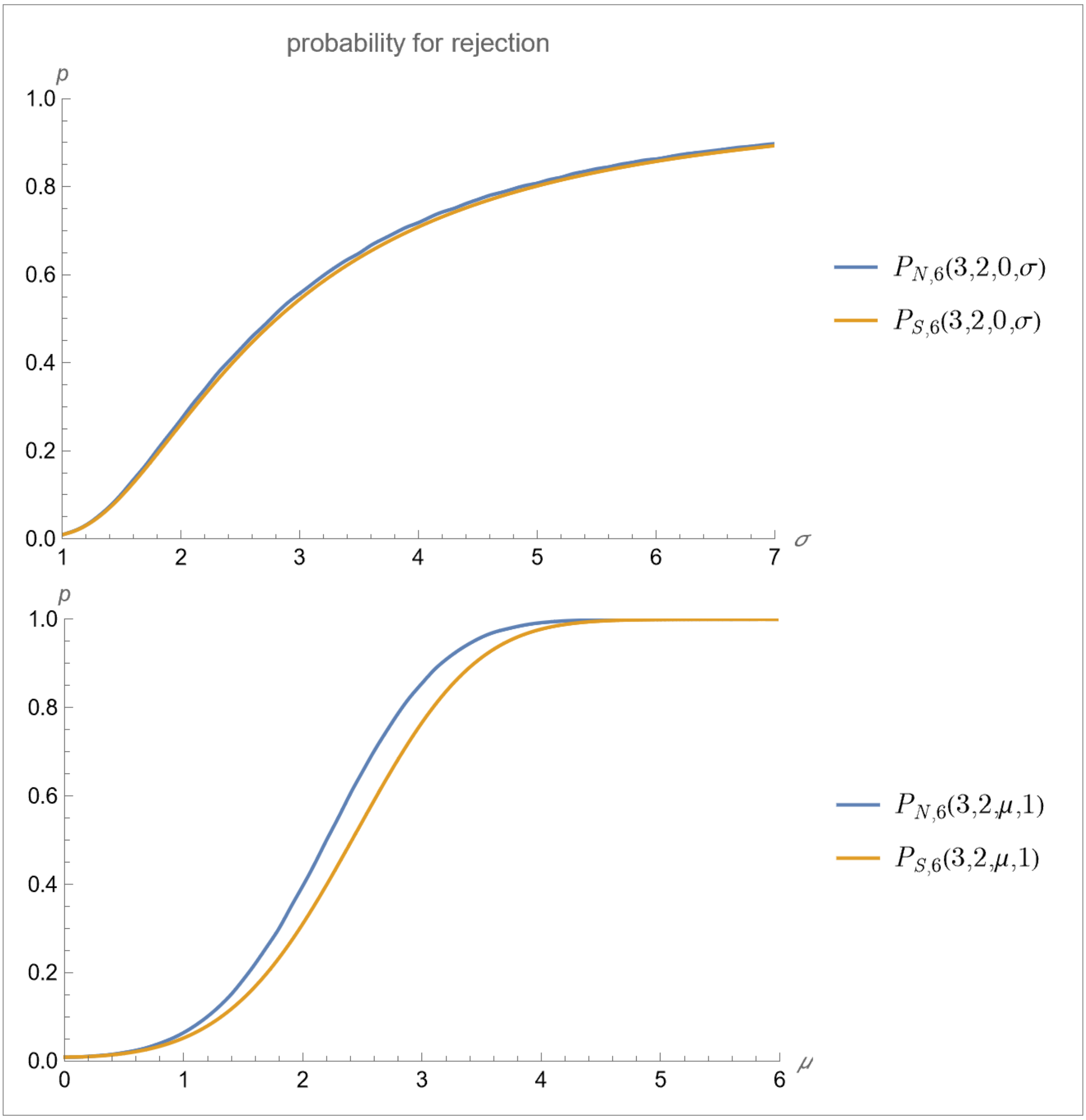

Fig 15: $P_{N,6}(3,2,\mu,\sigma)$ *vs* $P_{S,6}(3,2,\mu,\sigma)$

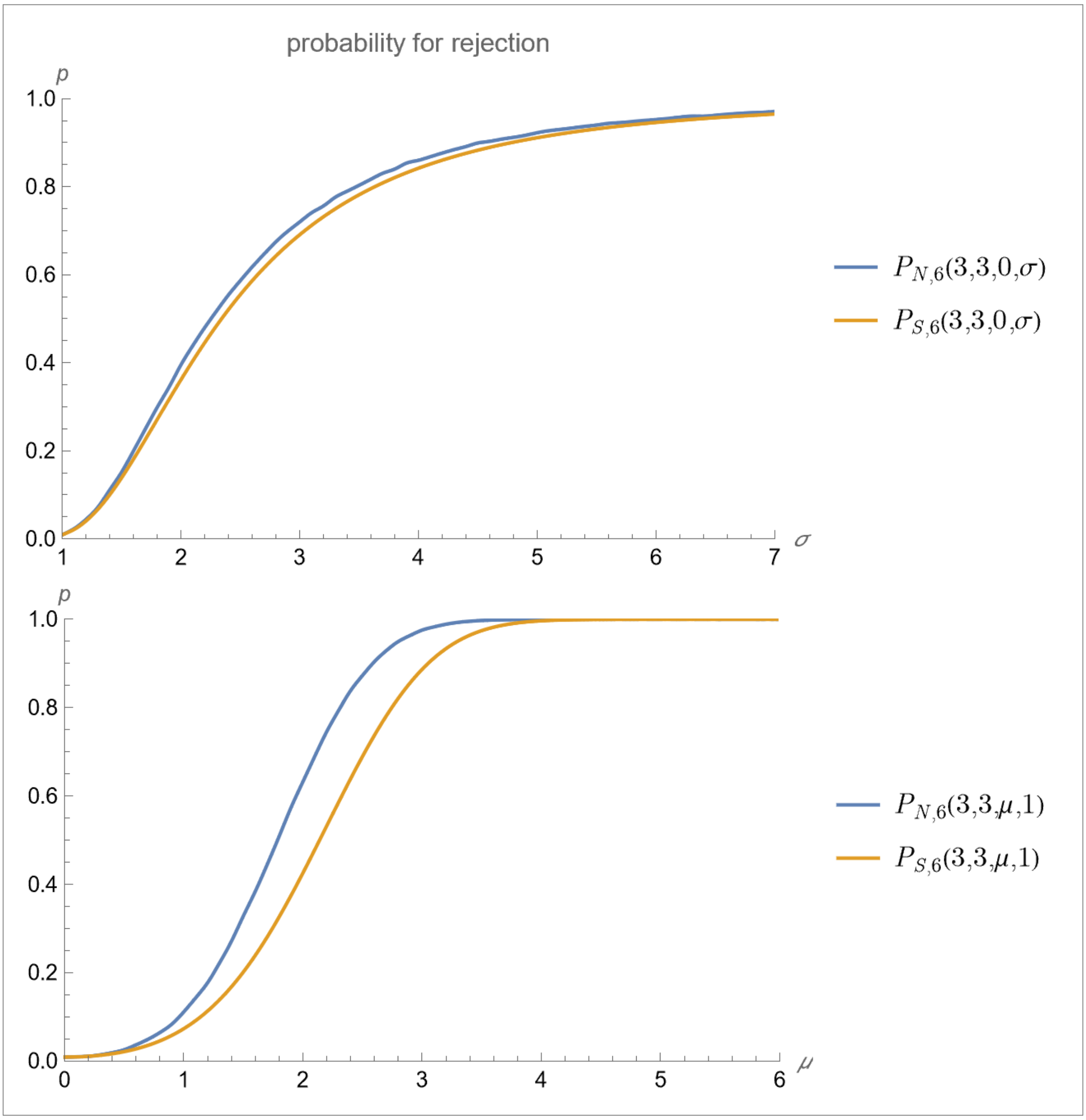

Fig 16: $P_{N,6}(3,3,\mu,\sigma)$ *vs* $P_{S,6}(3,3,\mu,\sigma)$

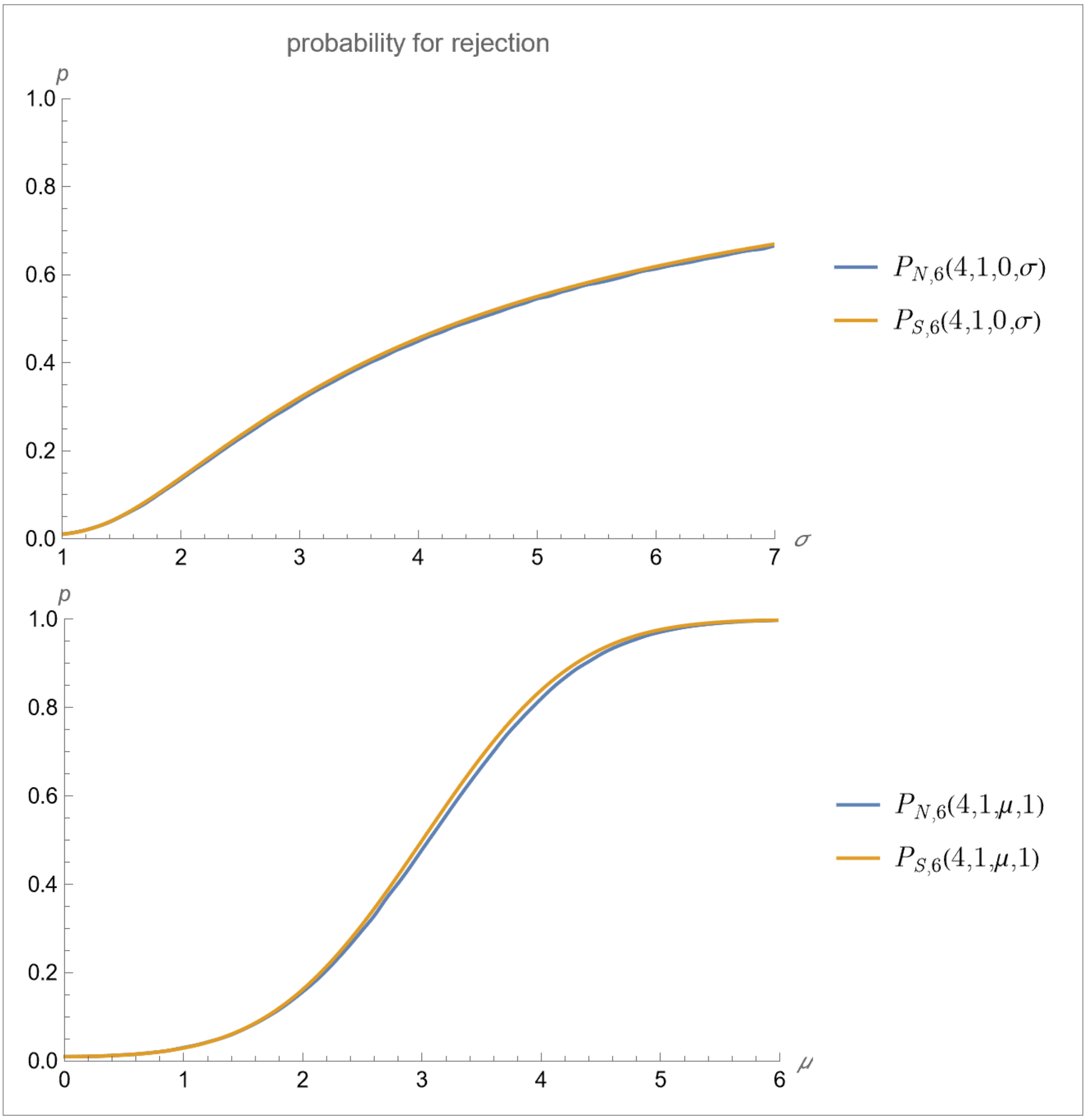

Fig 17: $P_{N,6}(4,1,\mu,\sigma)$ *vs* $P_{S,6}(4,1,\mu,\sigma)$

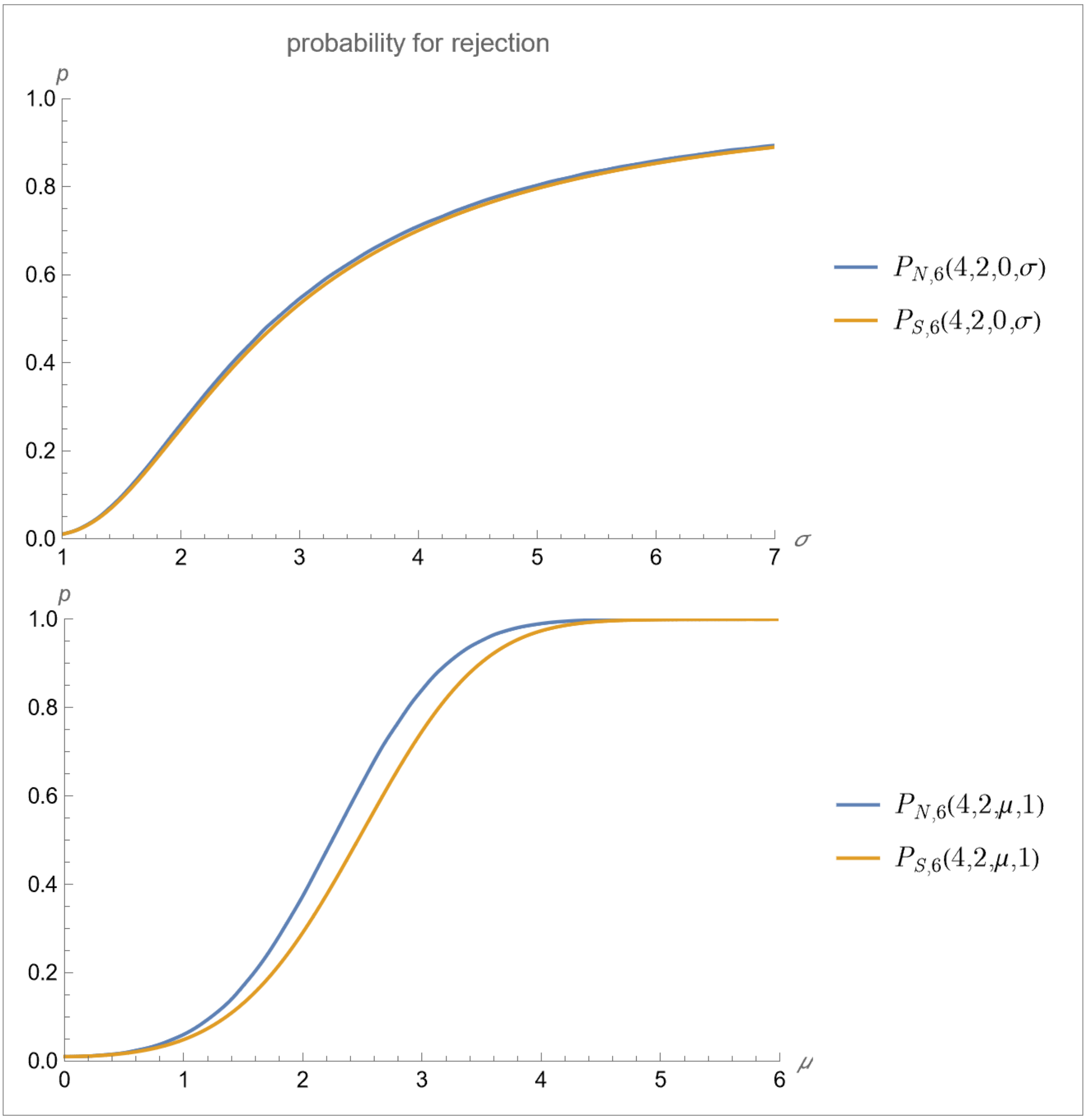

Fig 18: $P_{N,6}(4,2,\mu,\sigma)$ *vs* $P_{S,6}(4,2,\mu,\sigma)$

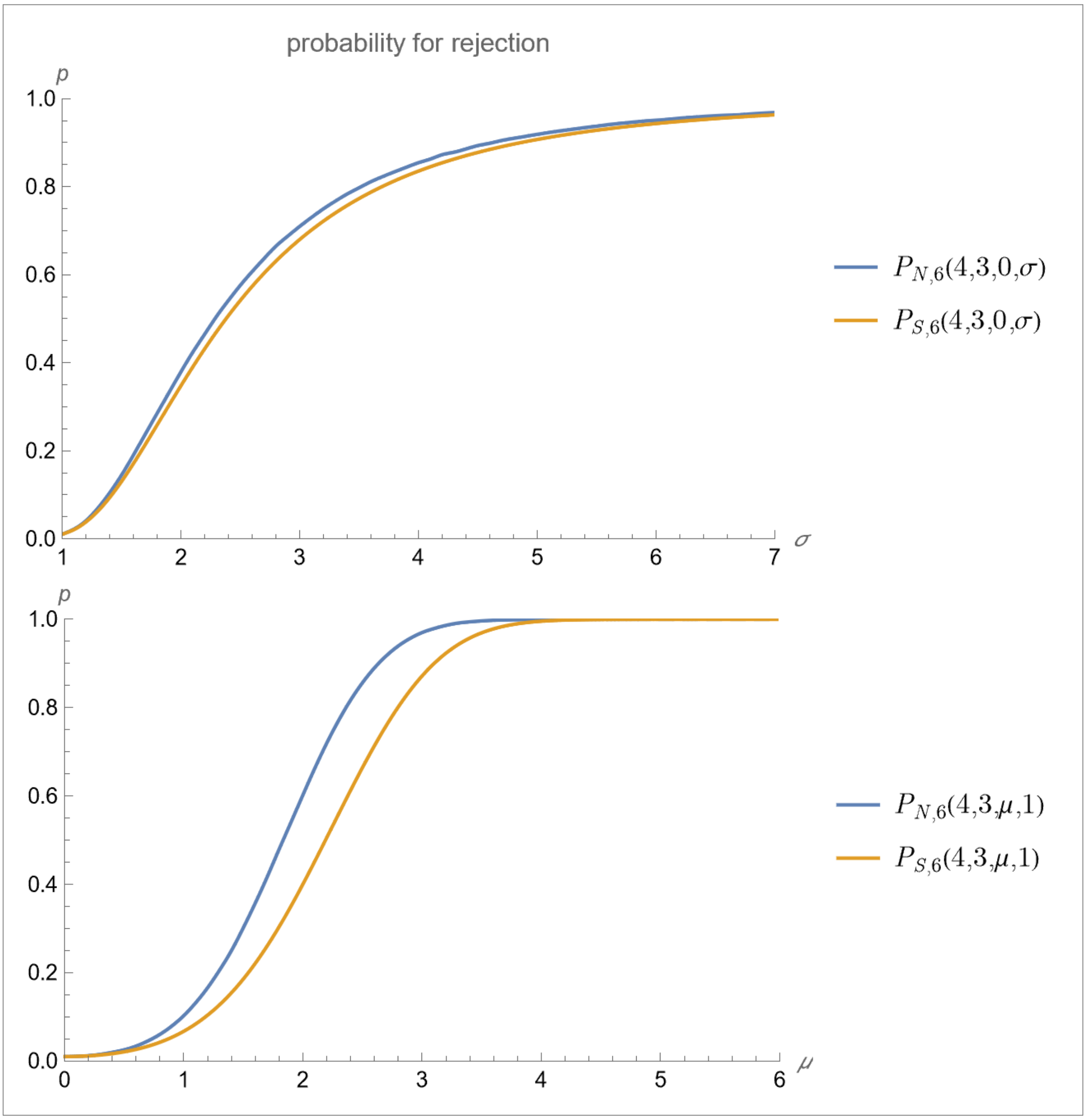

Fig 19: $P_{N,6}(4,3,\mu,\sigma)$ *vs* $P_{S,6}(4,3,\mu,\sigma)$

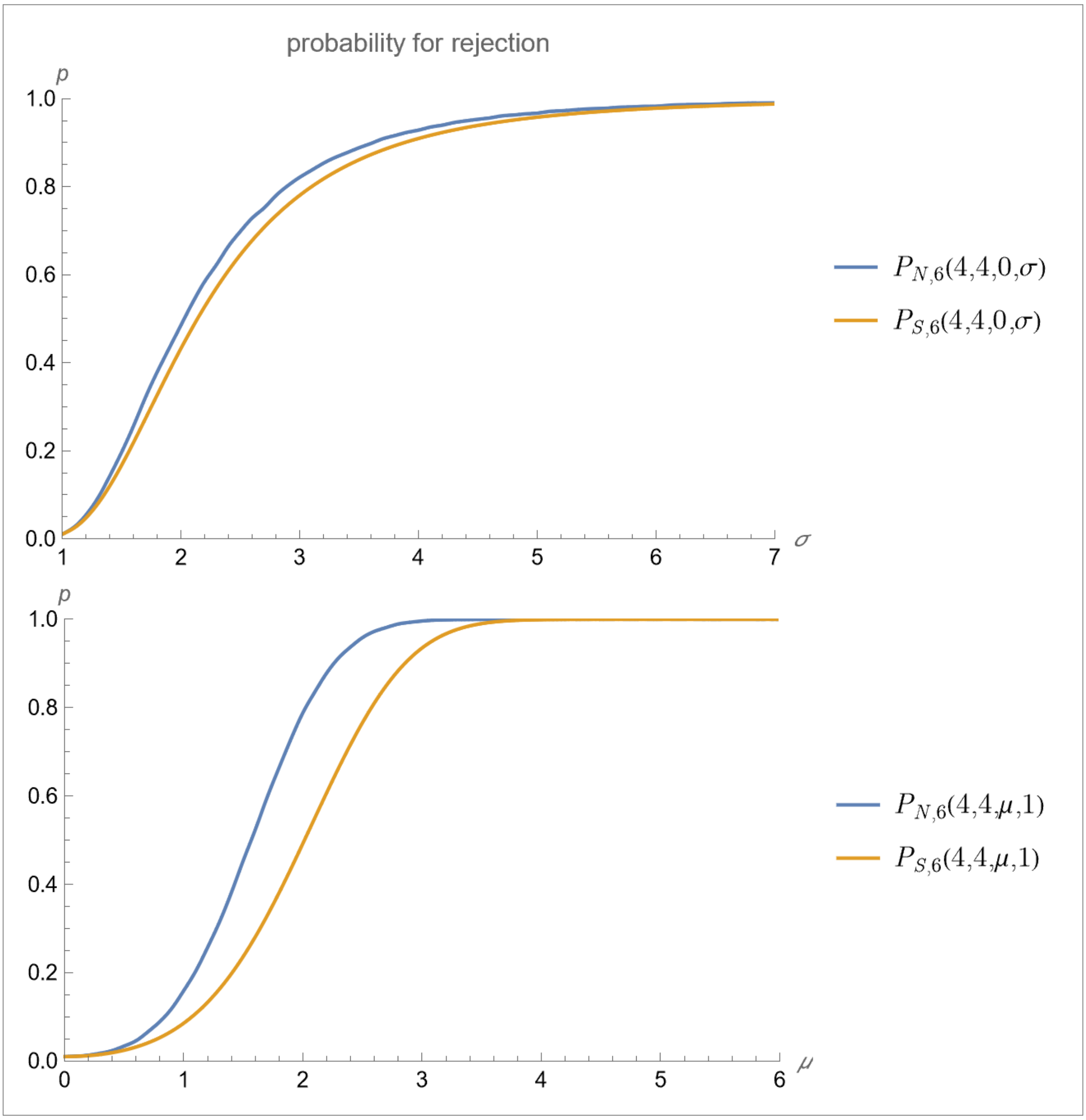

Fig 20: $P_{N,6}(4,4,\mu,\sigma)$ *vs* $P_{S,6}(4,4,\mu,\sigma)$

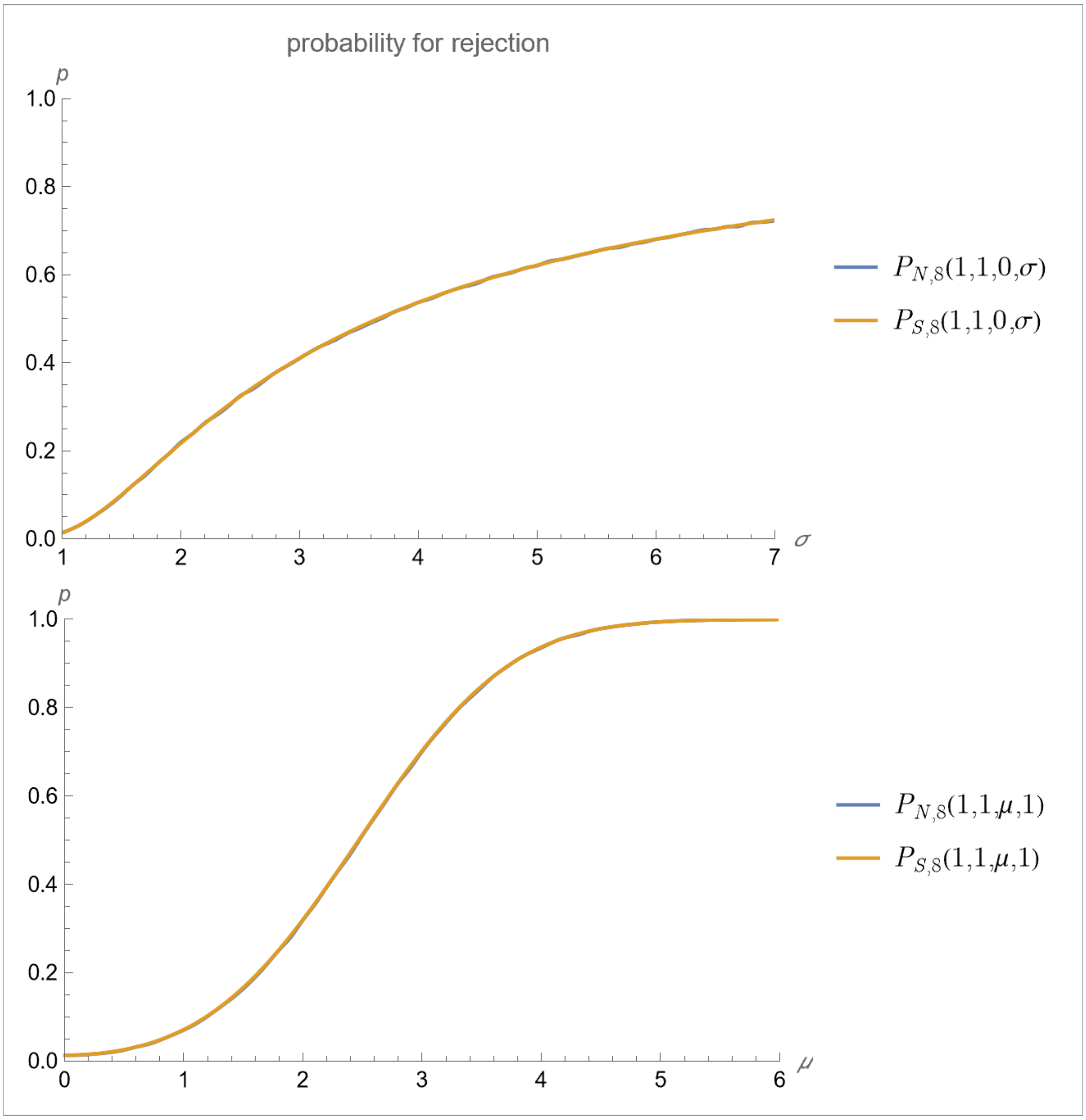

Fig 21: $P_{N,8}(1,1,\mu,\sigma)$ *vs* $P_{S,8}(1,1,\mu,\sigma)$

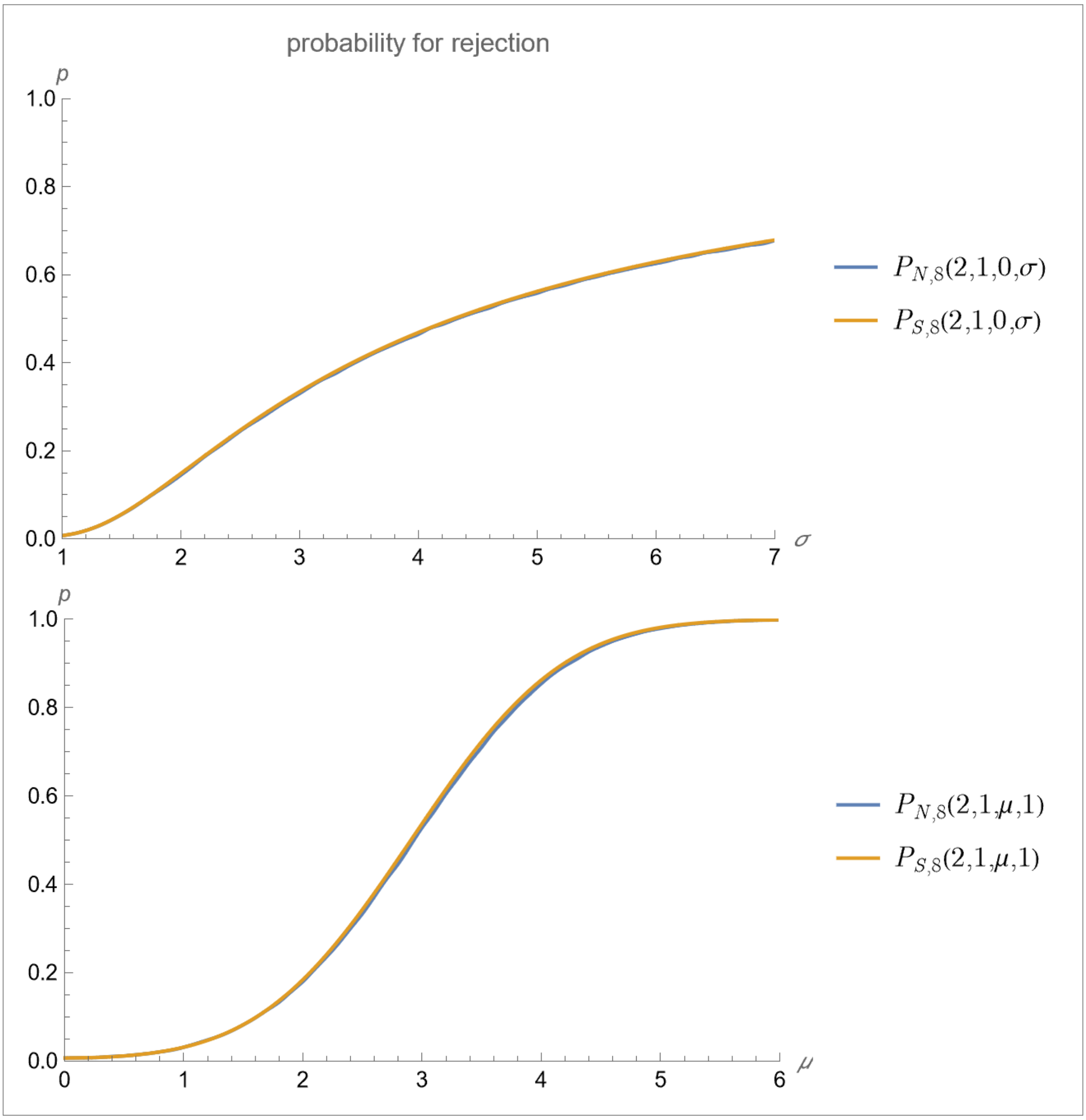

Fig 22: $P_{N,8}(2,1,\mu,\sigma)$ *vs* $P_{S,8}(2,1,\mu,\sigma)$

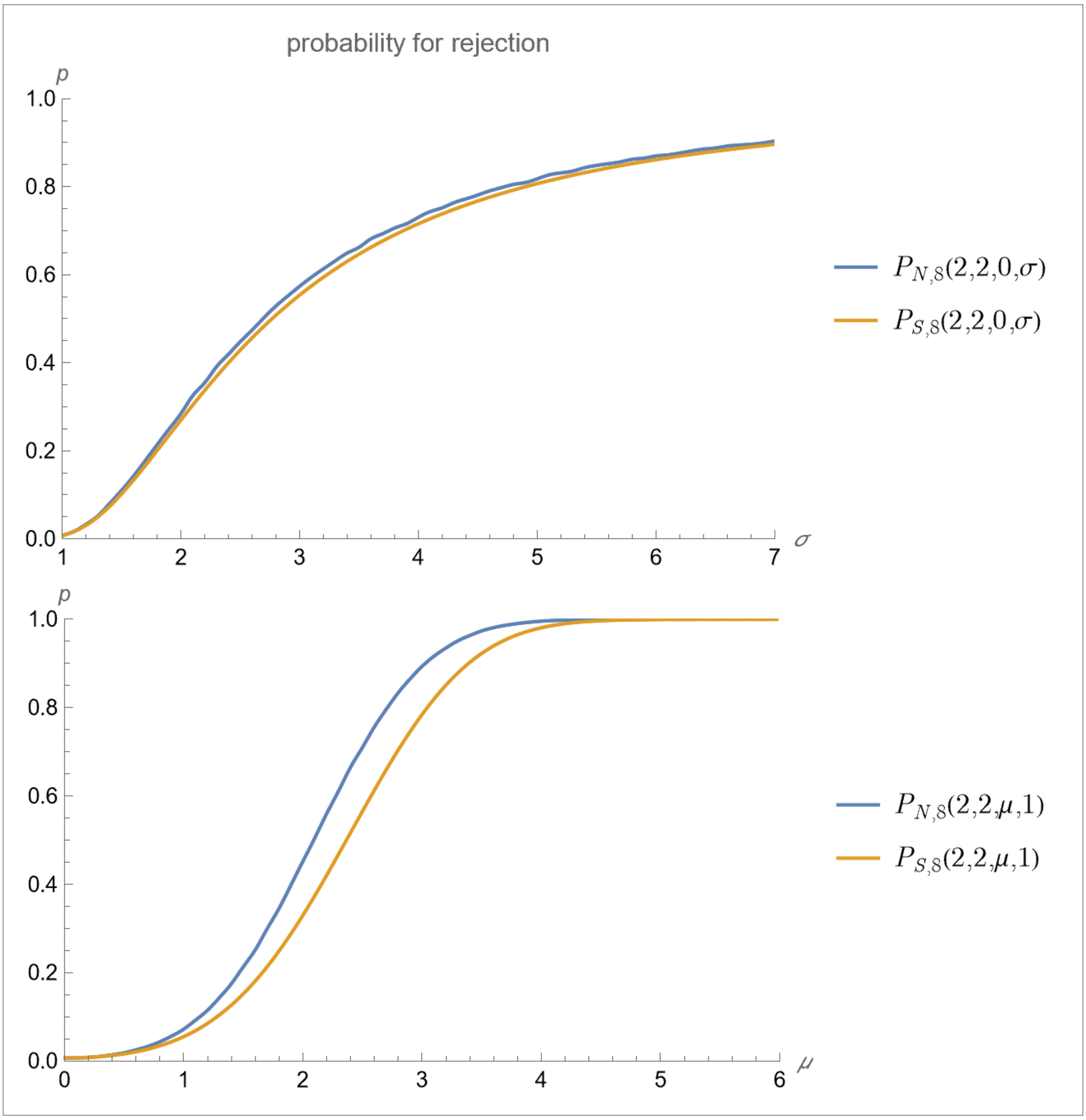

Fig 23: $P_{N,8}(2,2,\mu,\sigma)$ *vs* $P_{S,8}(2,2,\mu,\sigma)$

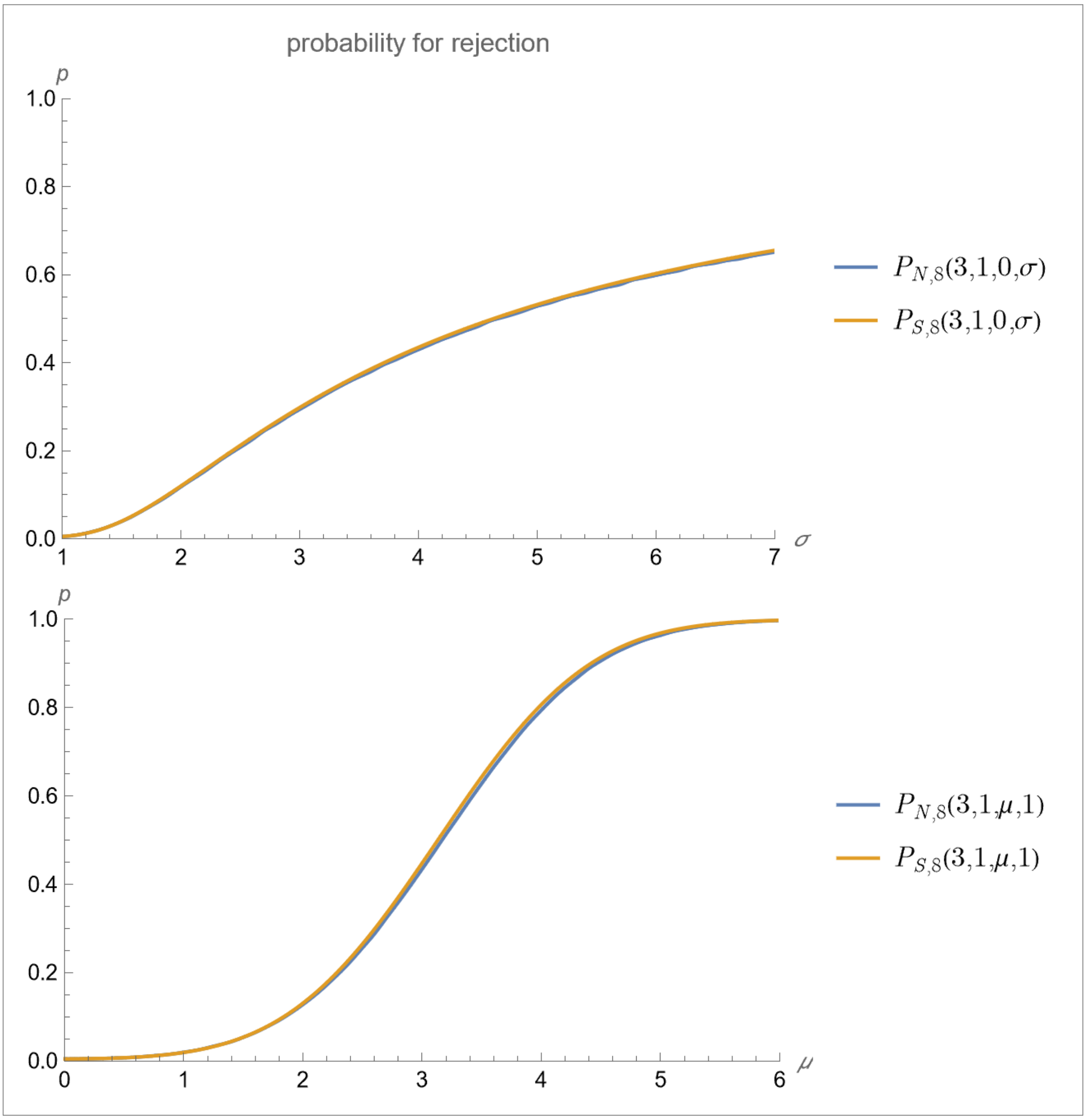

Fig 24: $P_{N,8}(3,1,\mu,\sigma)$ *vs* $P_{S,8}(3,1,\mu,\sigma)$

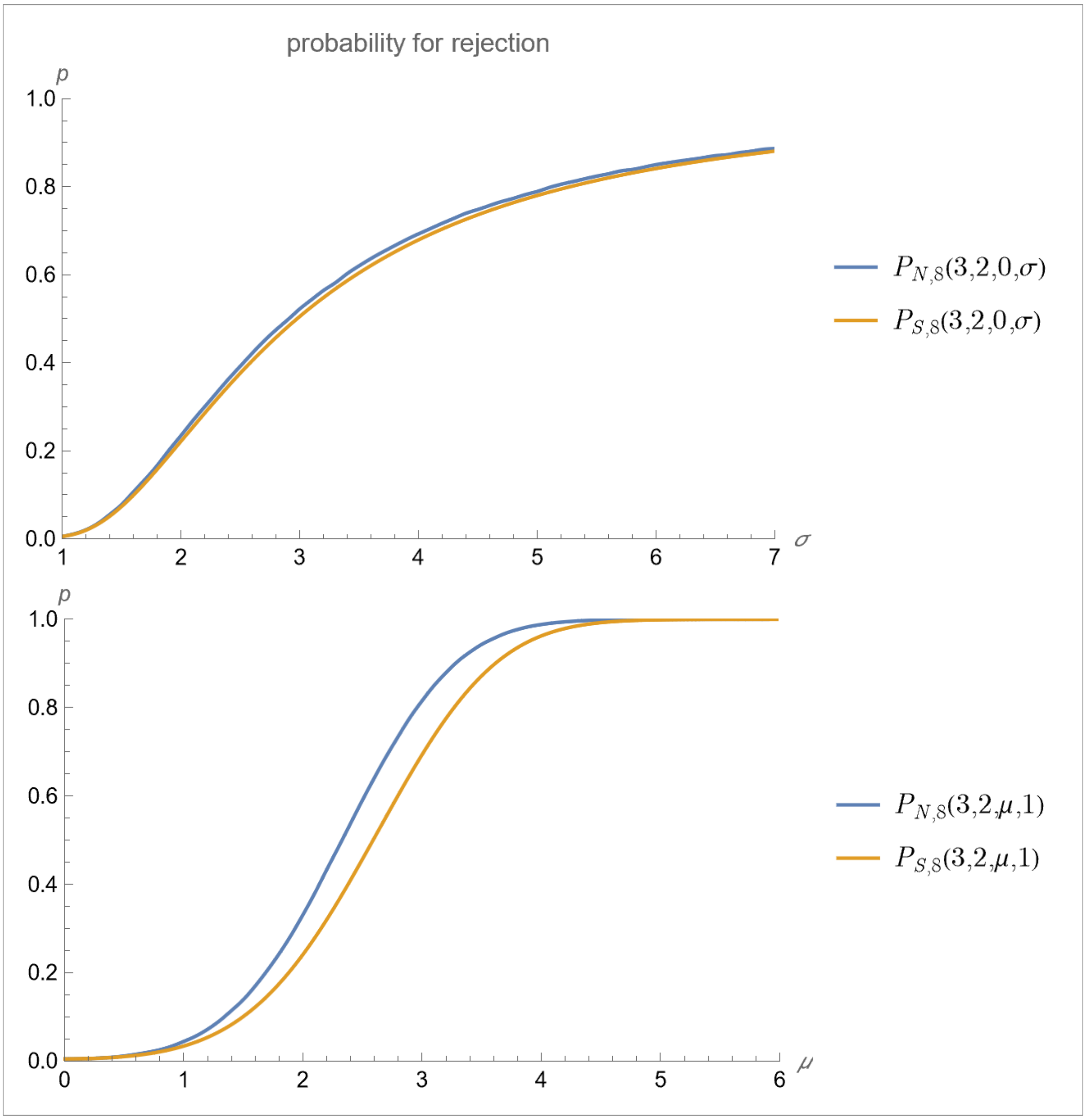

Fig 25: $P_{N,8}(3,2,\mu,\sigma)$ *vs* $P_{S,8}(3,2,\mu,\sigma)$

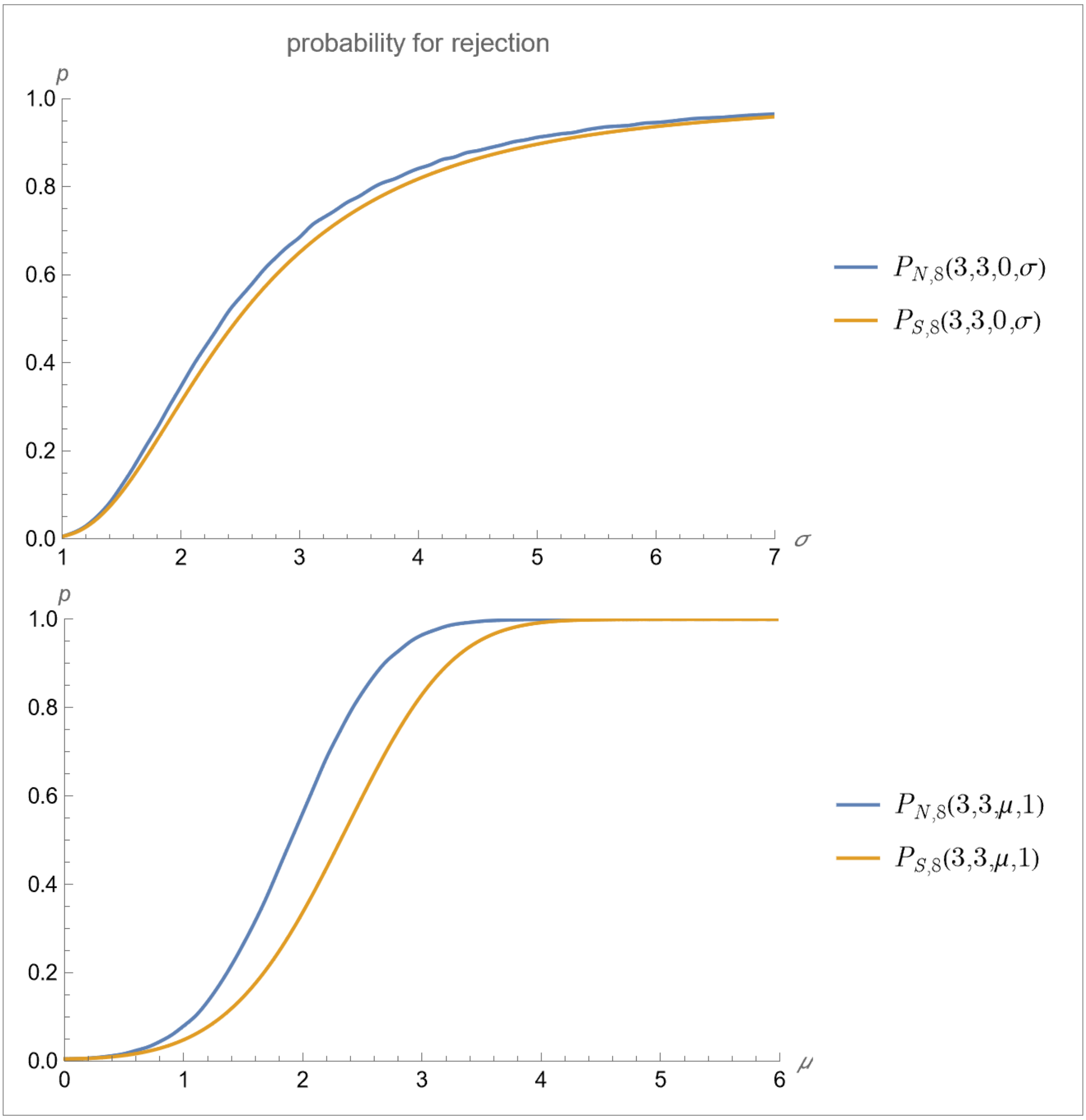

Fig 26: $P_{N,8}(3,3,\mu,\sigma)$ *vs* $P_{S,8}(3,3,\mu,\sigma)$

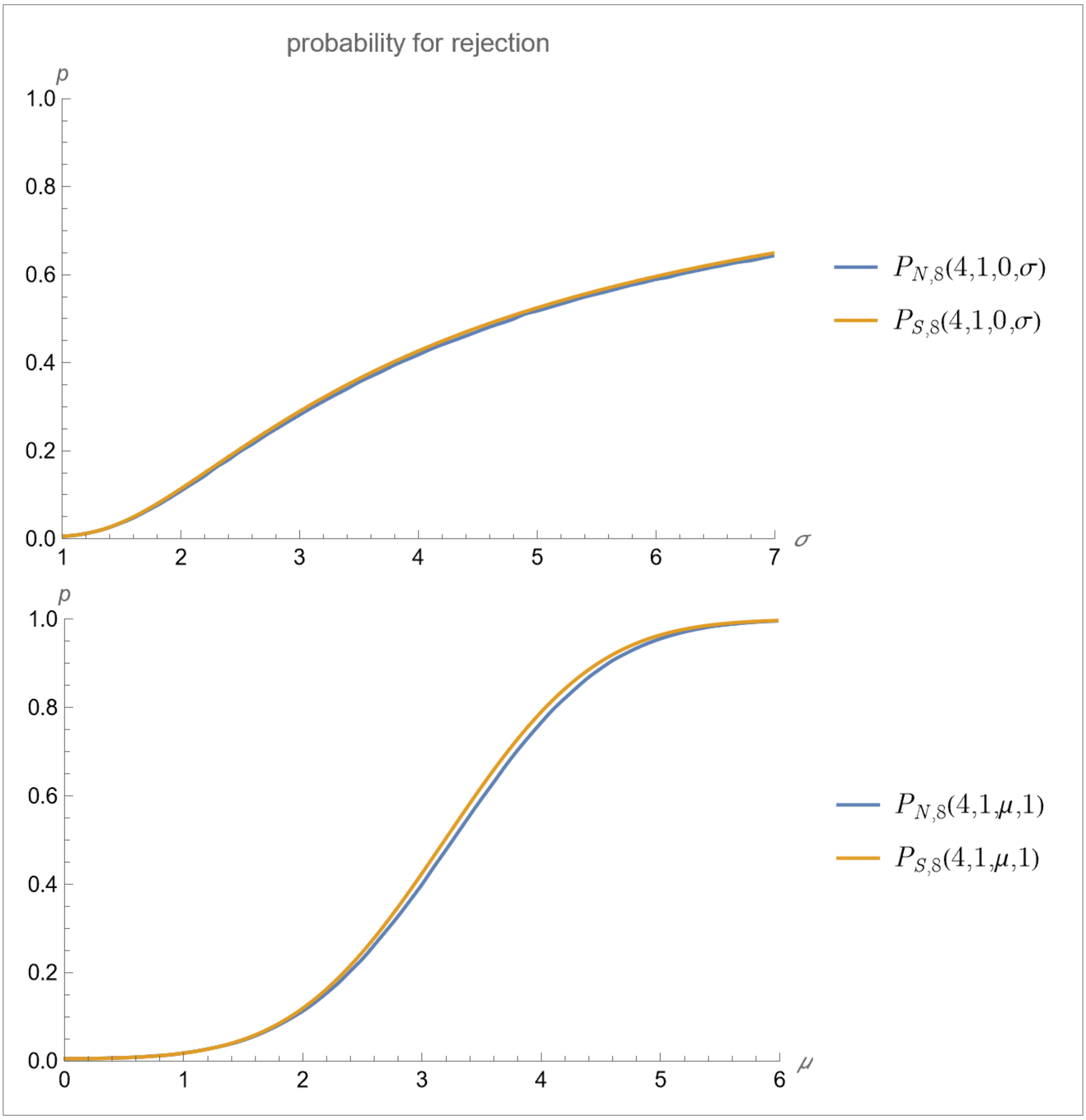

Fig 27: $P_{N,8}(4,1,\mu,\sigma)$ *vs* $P_{S,8}(4,1,\mu,\sigma)$

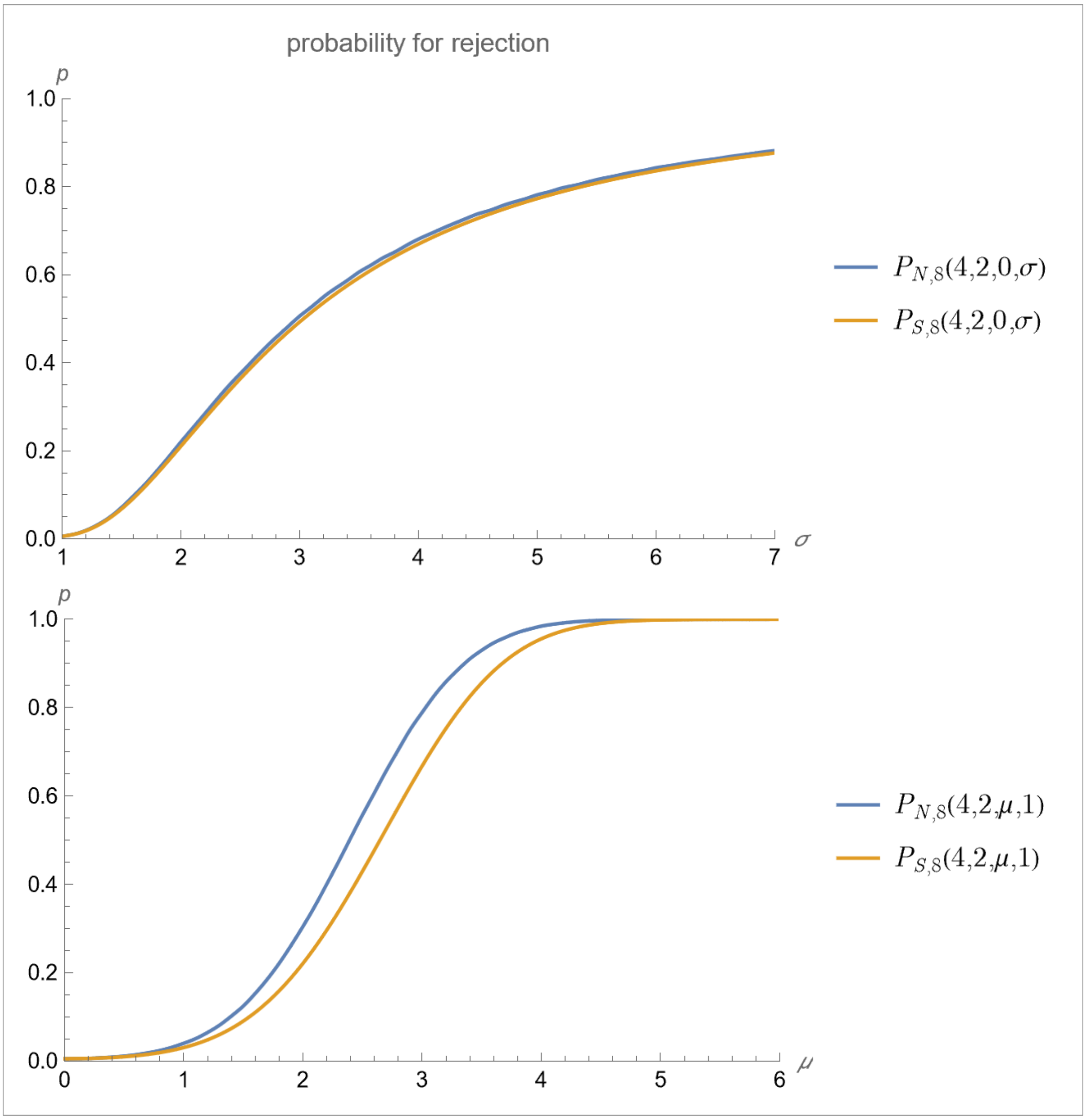

Fig 28: $P_{N,8}(4,2,\mu,\sigma)$ *vs* $P_{S,8}(4,2,\mu,\sigma)$

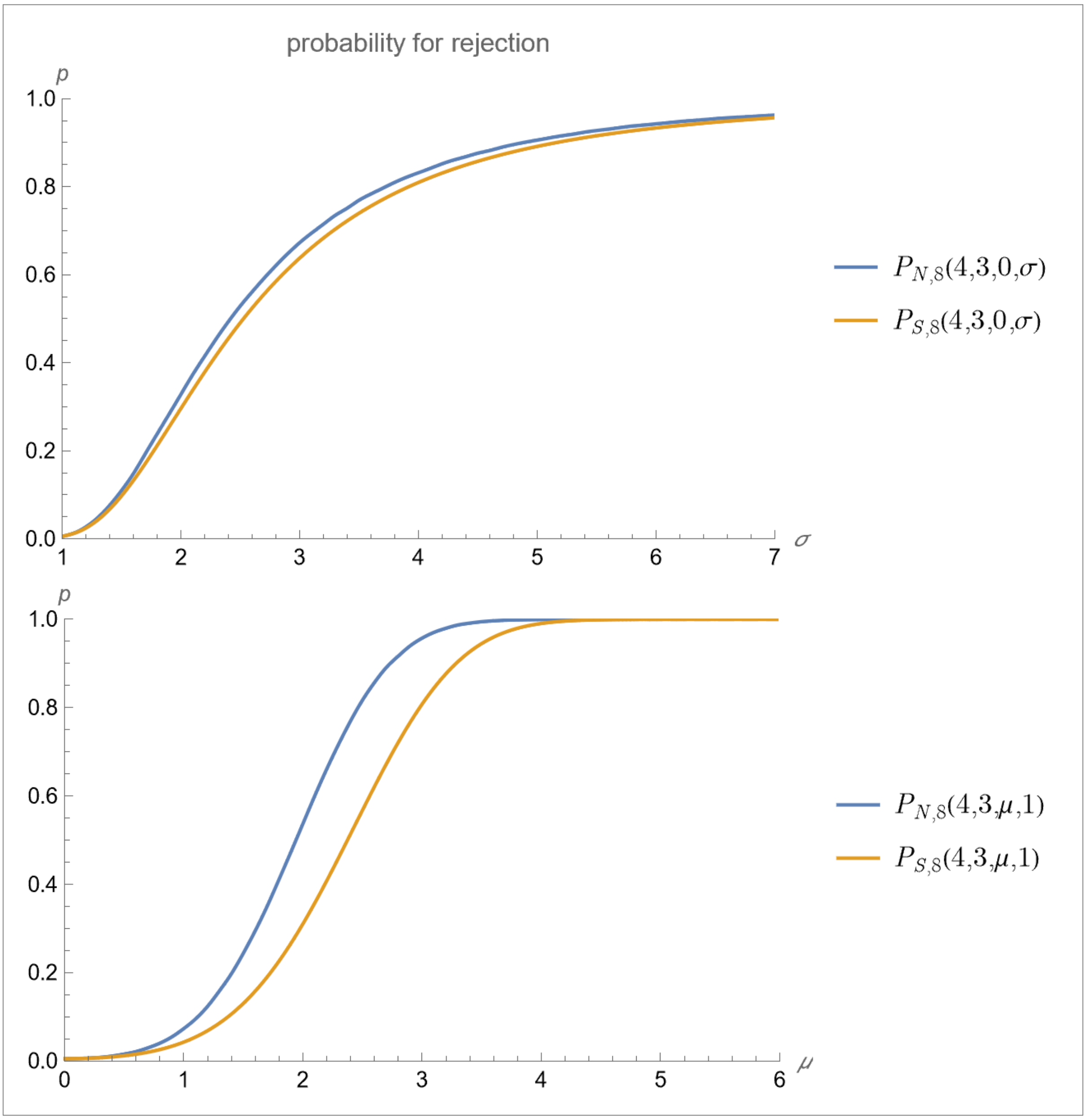

Fig 29: $P_{N,8}(4,3,\mu,\sigma)$ *vs* $P_{S,8}(4,3,\mu,\sigma)$

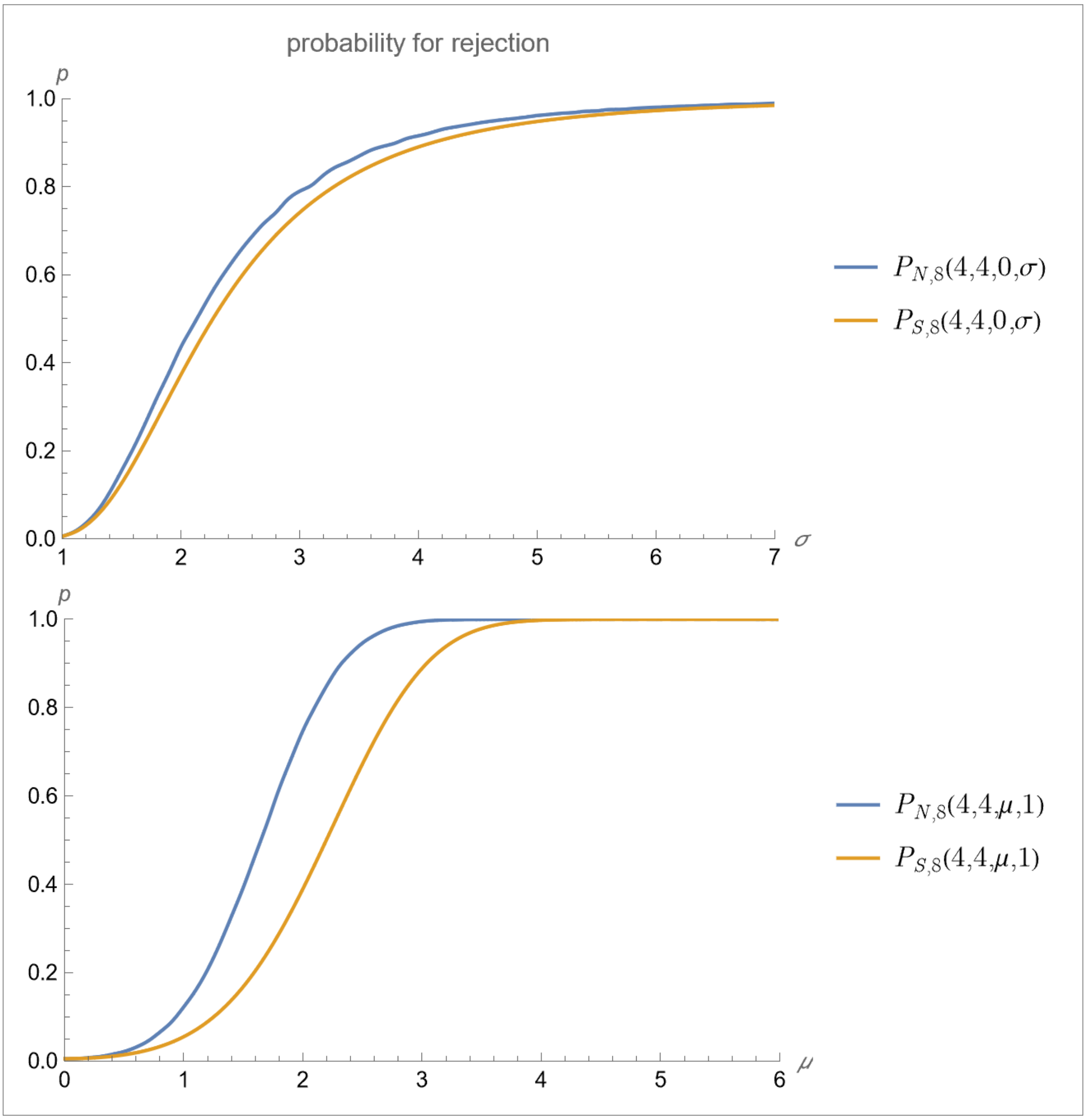

Fig 30: $P_{N,8}(4,4,\mu,\sigma)$ *vs* $P_{S,8}(4,4,\mu,\sigma)$

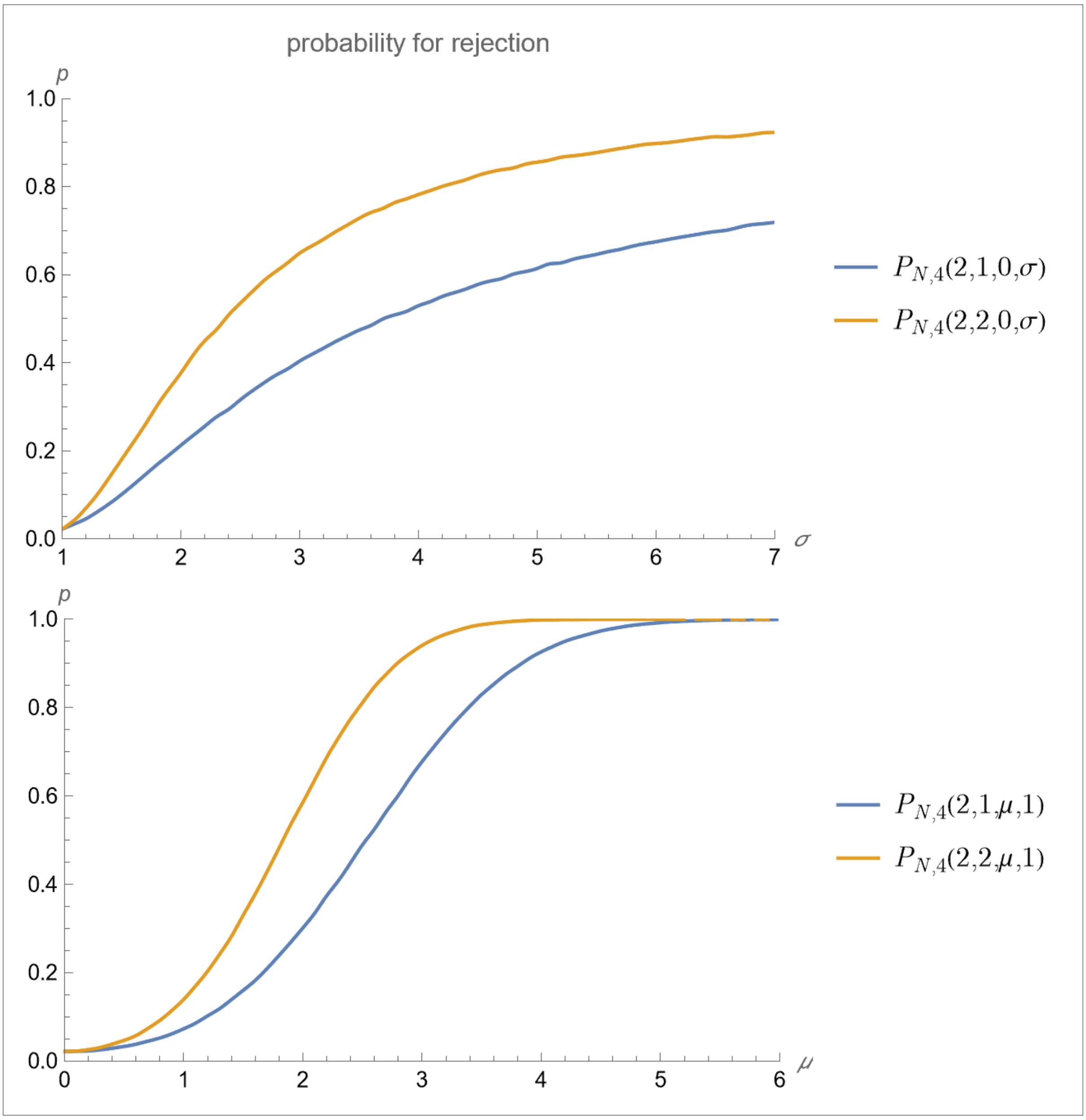

Fig 31: $P_{N,4}(2{,}1,\mu,\sigma)$ *vs* $P_{N,4}(2{,}2,\mu,\sigma)$

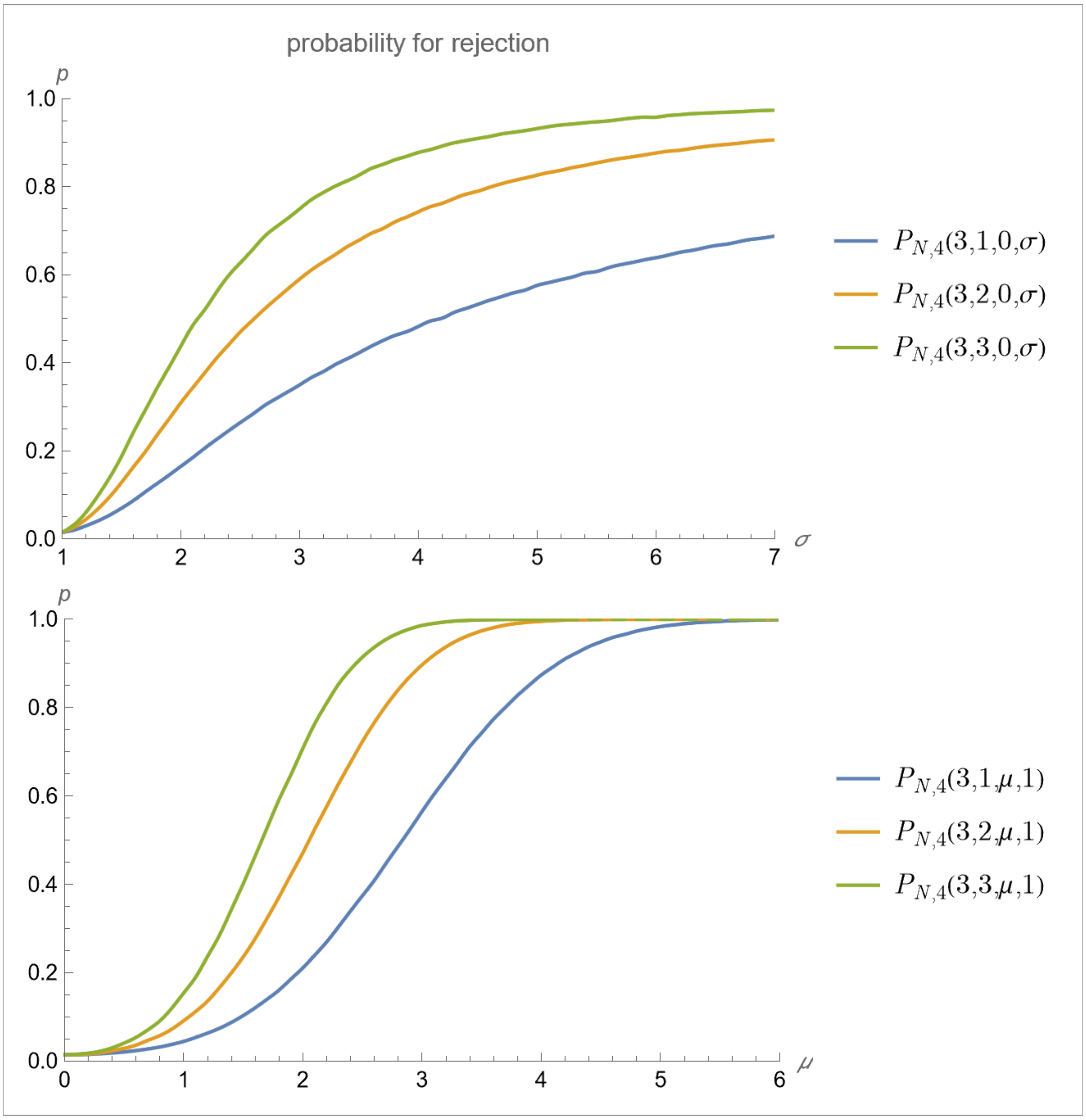

Fig 32: $P_{N,4}(3{,}1,\mu,\sigma)$ *vs* $P_{N,4}(3{,}2,\mu,\sigma)$ *vs* $P_{N,4}(3{,}3,\mu,\sigma)$

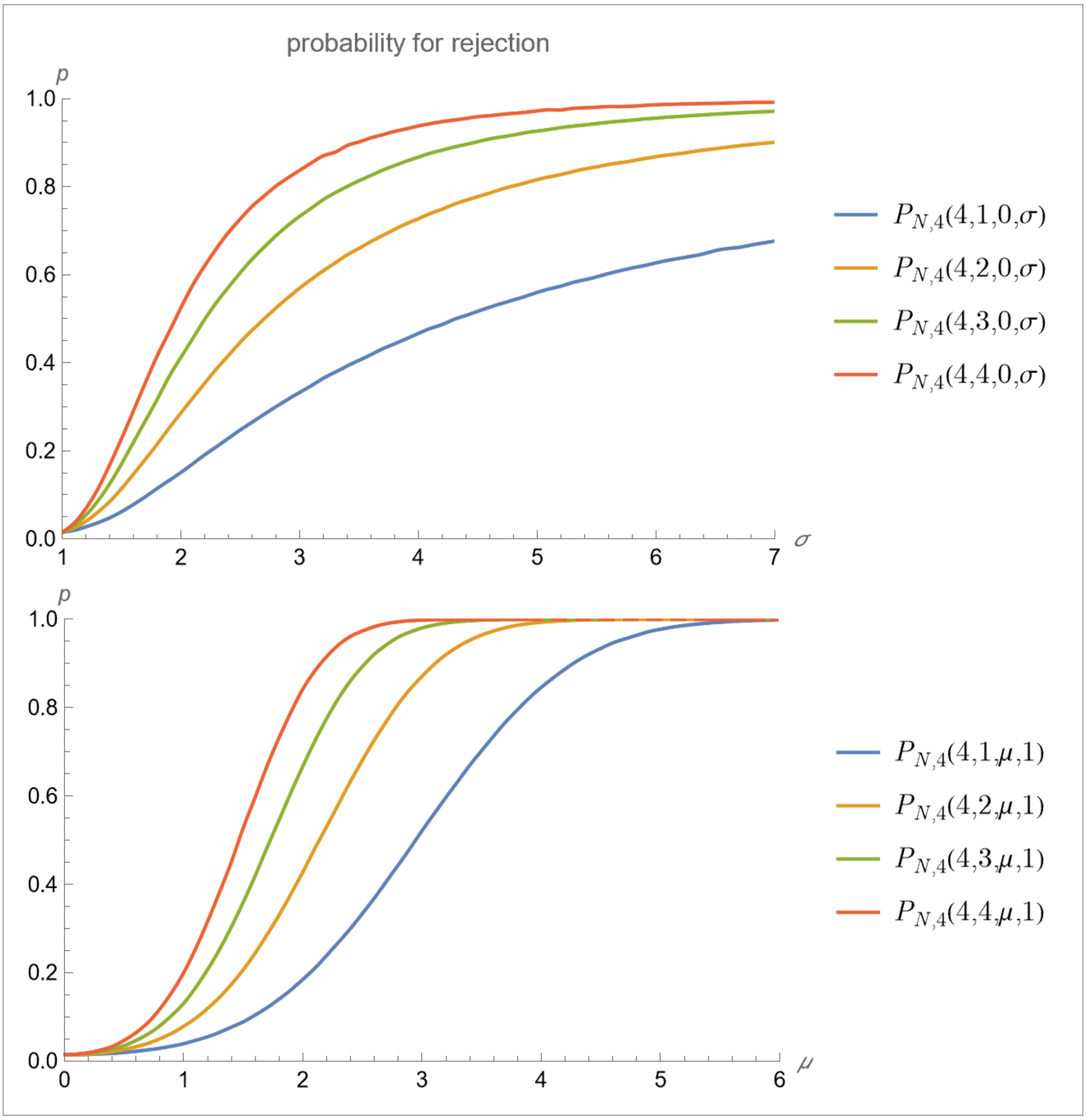

Fig 33: $P_{N,4}(4,1,\mu,\sigma)$ *vs* $P_{N,4}(4,2,\mu,\sigma)$ *vs* $P_{N,4}(4,3,\mu,\sigma)$ *vs* $P_{N,4}(4,4,\mu,\sigma)$

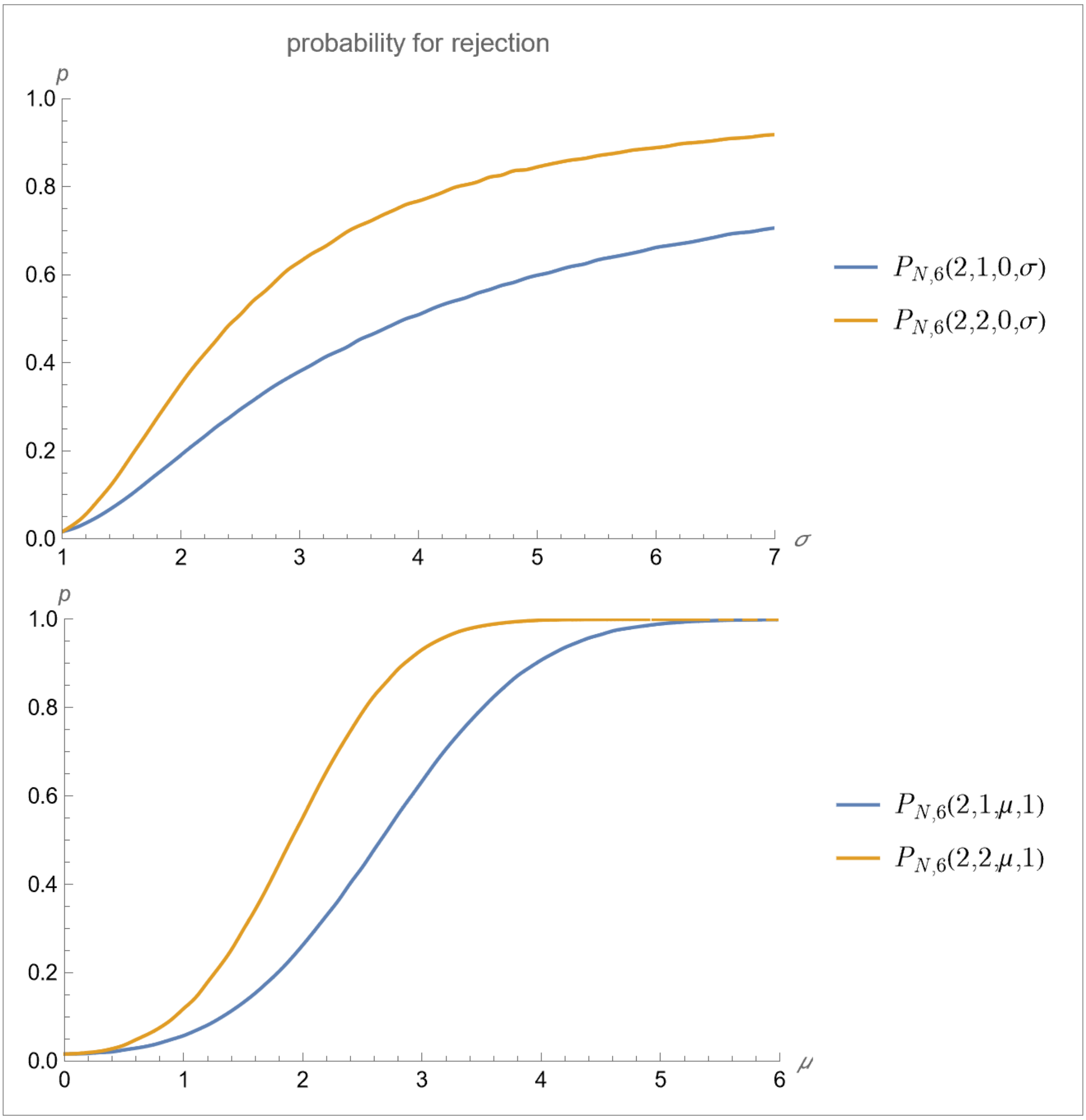

Fig 34: $P_{N,6}(2,1,\mu,\sigma)$ *vs* $P_{N,6}(2,2,\mu,\sigma)$

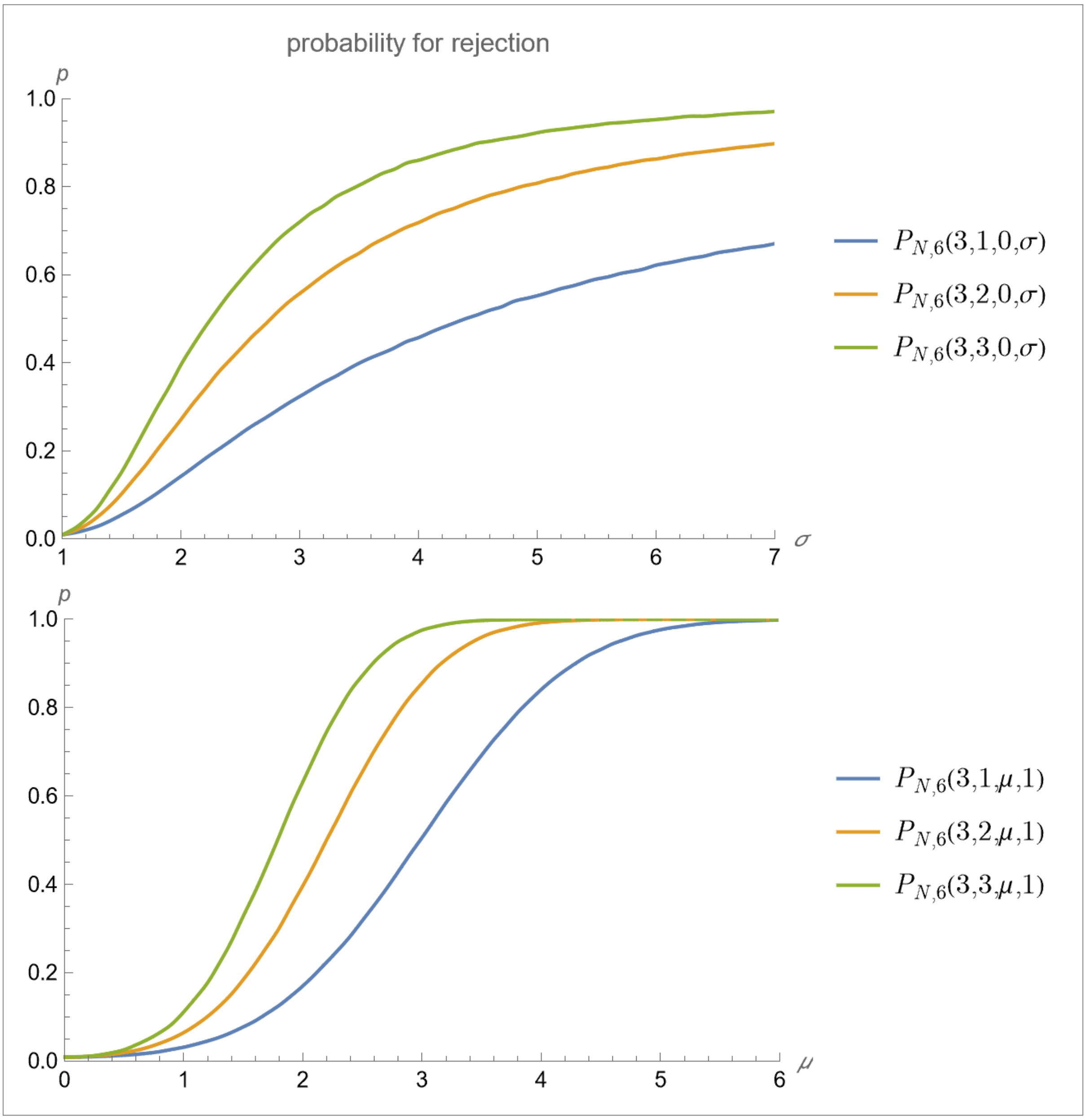

Fig 35: $P_{N,6}(3,1,\mu,\sigma)$ *vs* $P_{N,6}(3,2,\mu,\sigma)$ *vs* $P_{N,6}(3,3,\mu,\sigma)$

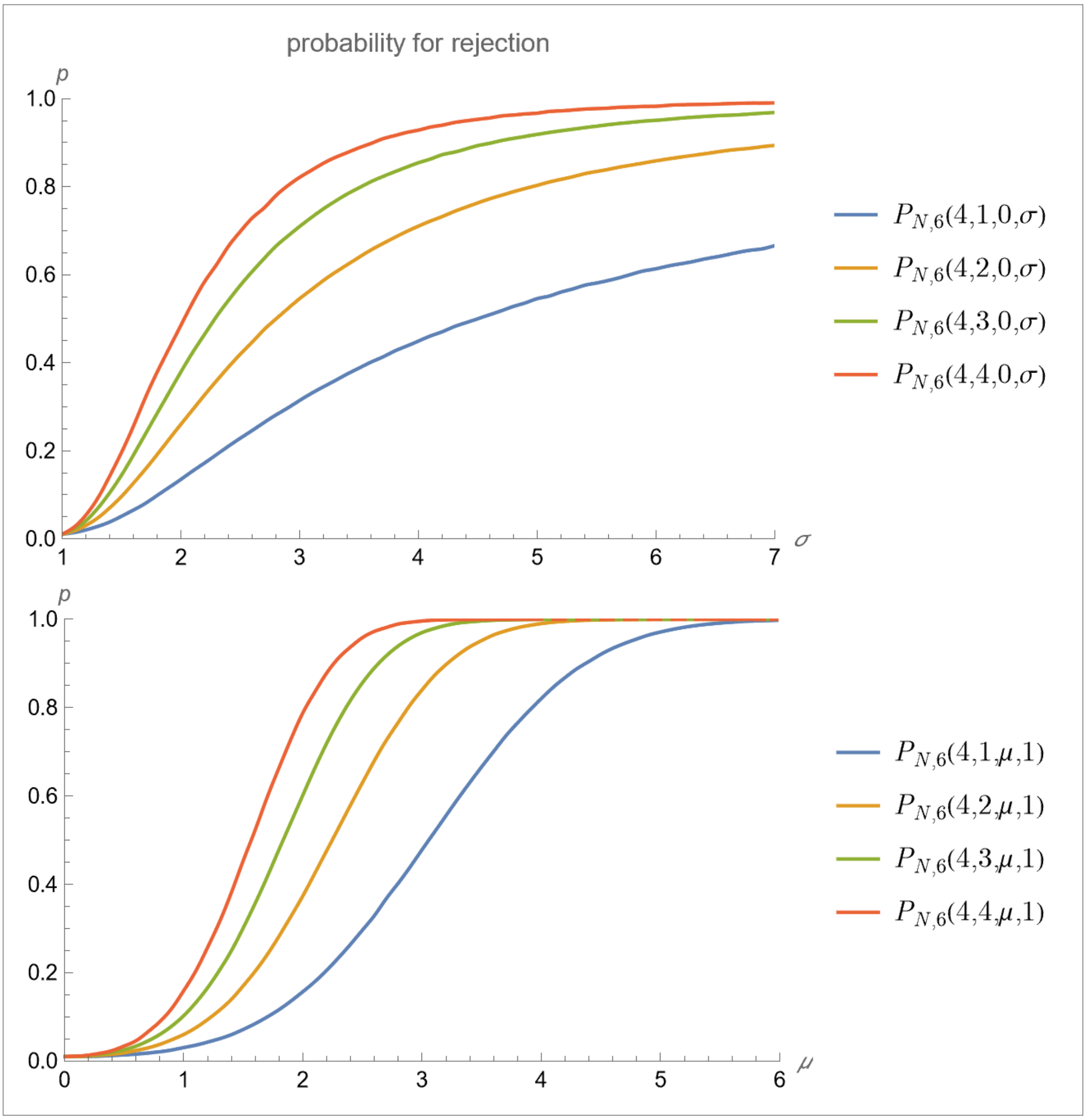

Fig 36: $P_{N,6}(4{,}1,\mu,\sigma)$ *vs* $P_{N,6}(4{,}2,\mu,\sigma)$ *vs* $P_{N,6}(4{,}3,\mu,\sigma)$ *vs* $P_{N,6}(4{,}4,\mu,\sigma)$

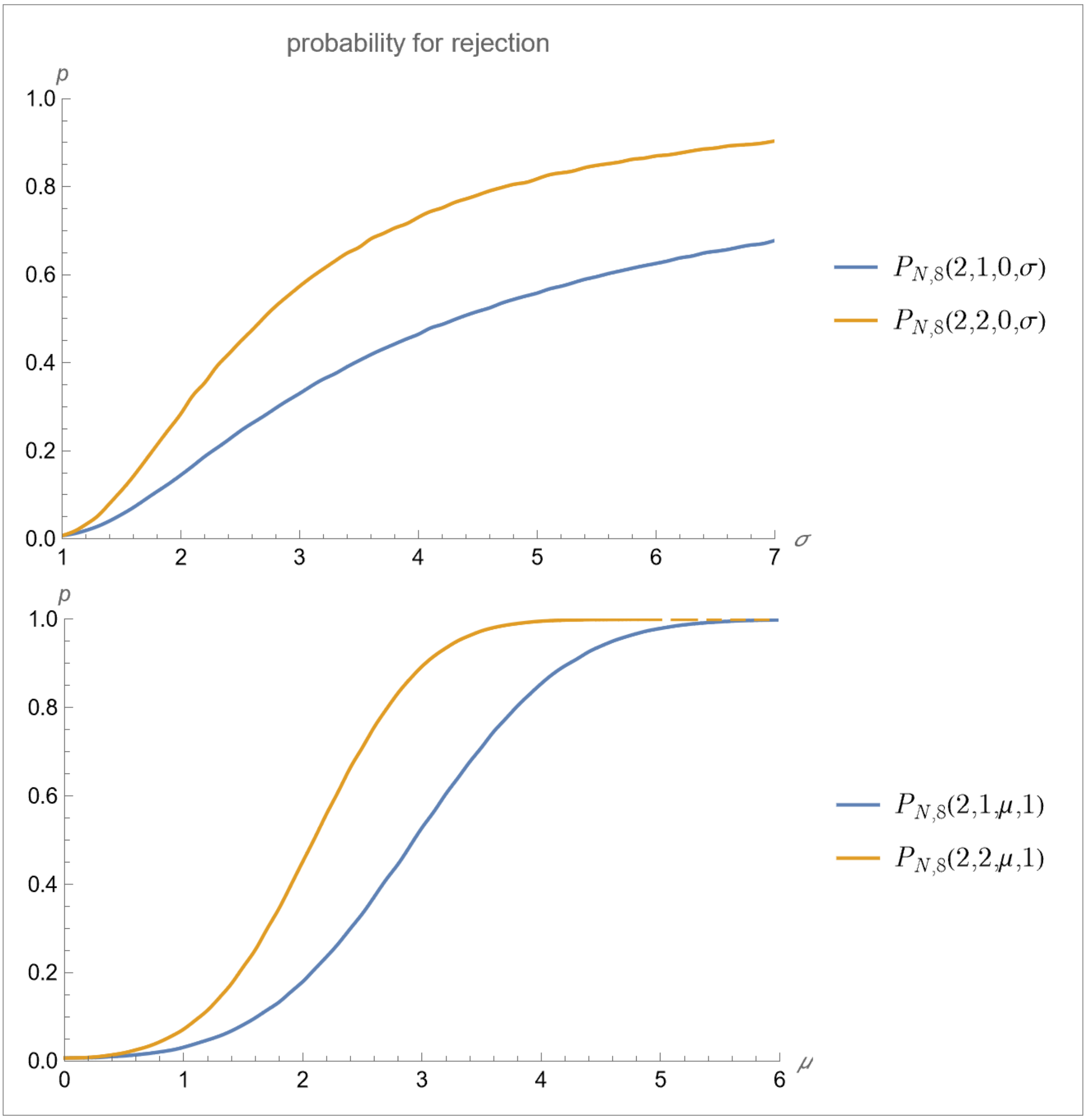

Fig 37: $P_{N,8}(2,1,\mu,\sigma)$ *vs* $P_{N,8}(2,2,\mu,\sigma)$

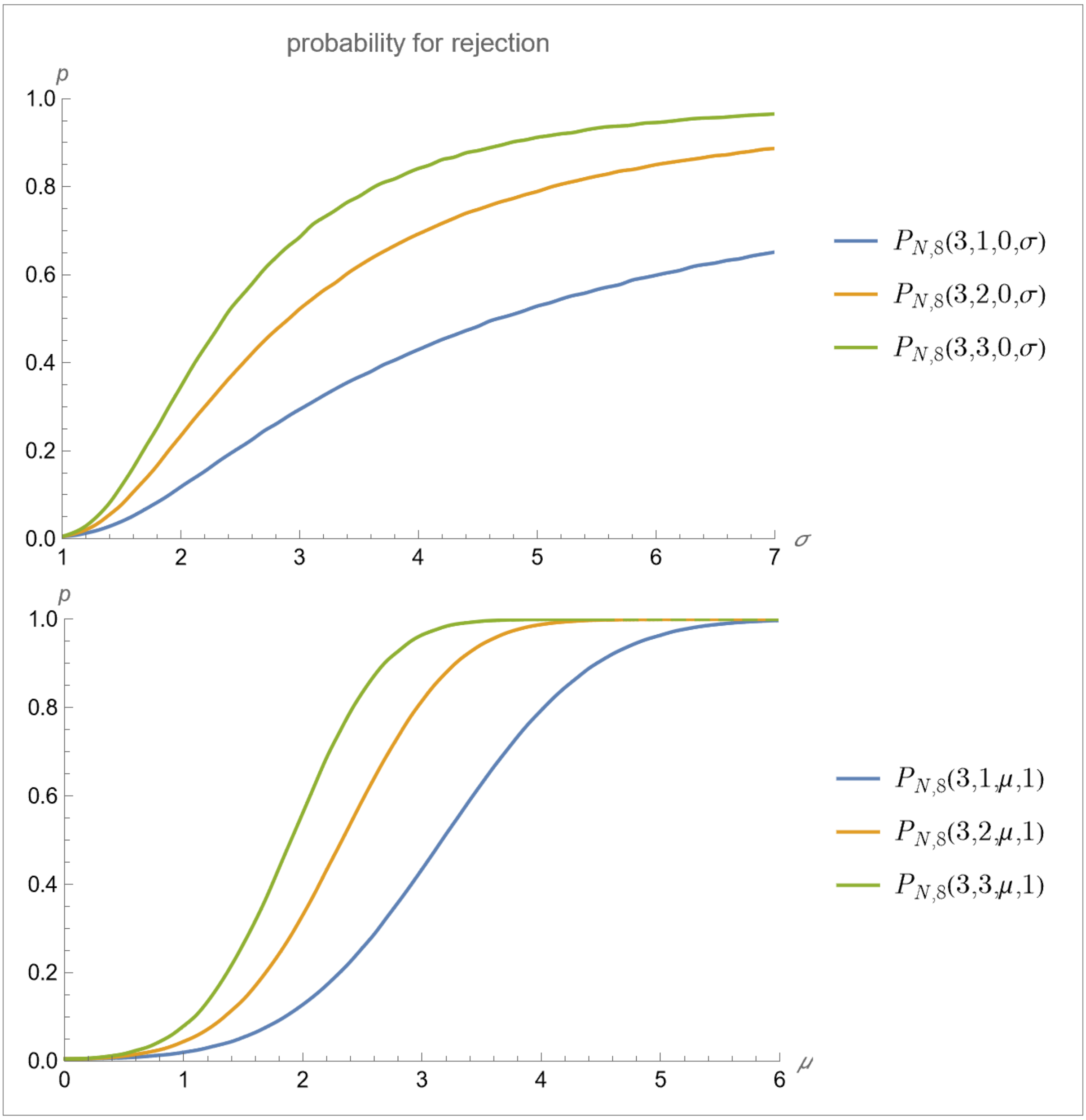

Fig 38: $P_{N,8}(3,1,\mu,\sigma)$ *vs* $P_{N,8}(3,2,\mu,\sigma)$ *vs* $P_{N,8}(3,3,\mu,\sigma)$

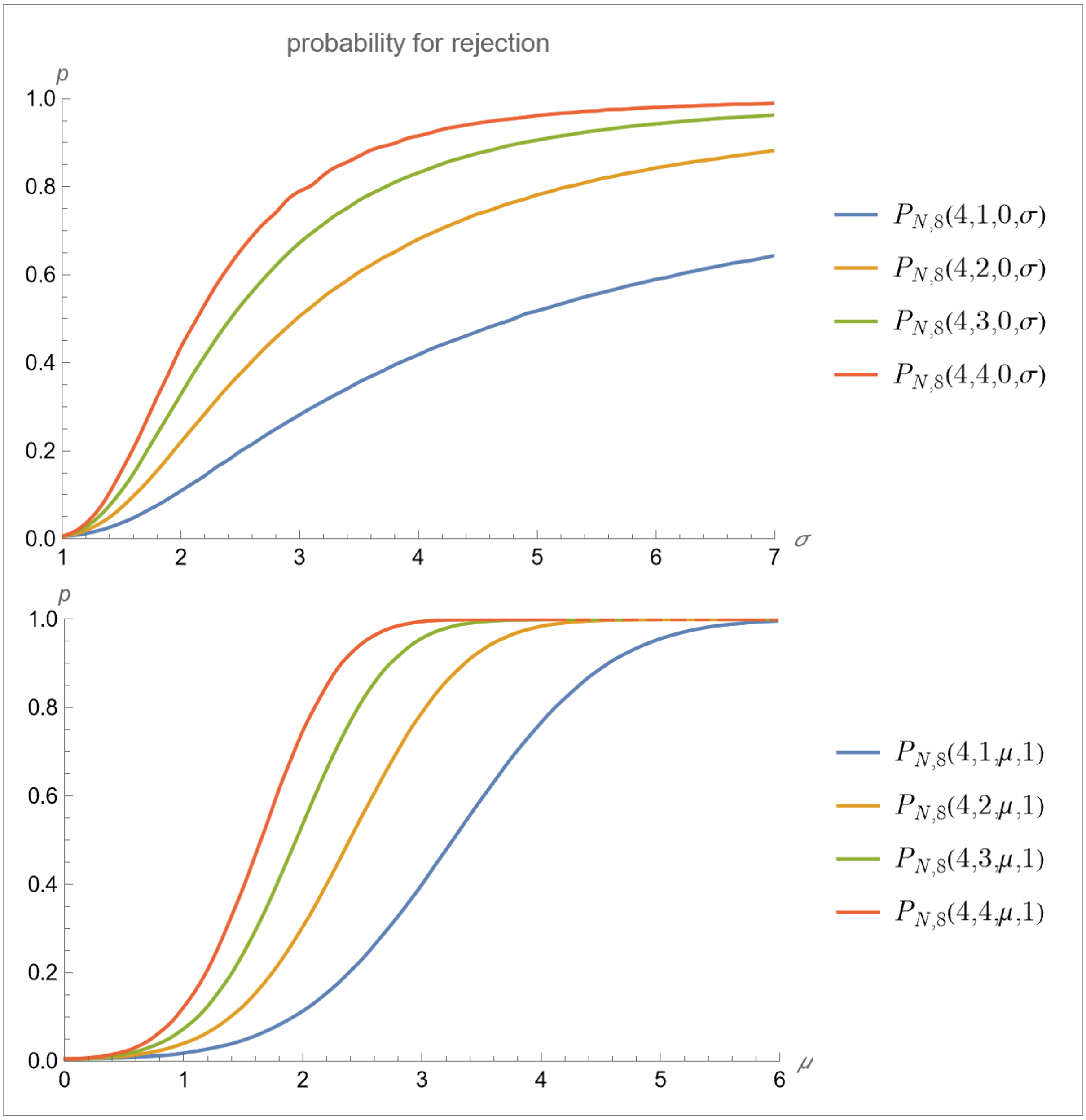

Fig 39: $P_{N,8}(4{,}1,\mu,\sigma)$ *vs* $P_{N,8}(4{,}2,\mu,\sigma)$ *vs* $P_{N,8}(4{,}3,\mu,\sigma)$ *vs* $P_{N,8}(4{,}4,\mu,\sigma)$

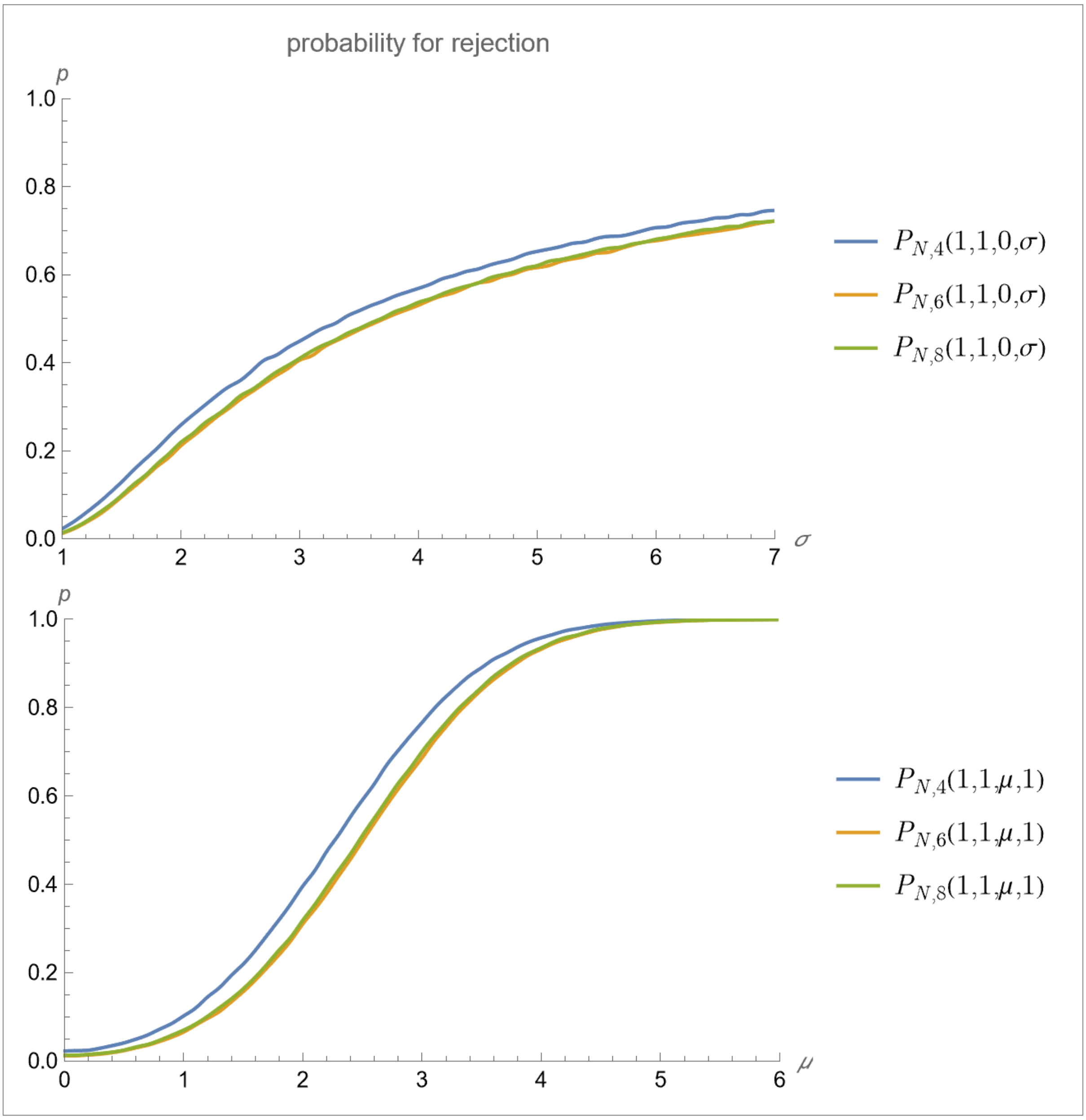

Fig 40: $P_{N,4}(1,1,\mu,\sigma)$ *vs* $P_{N,6}(1,1,\mu,\sigma)$ *vs* $P_{N,8}(1,1,\mu,\sigma)$

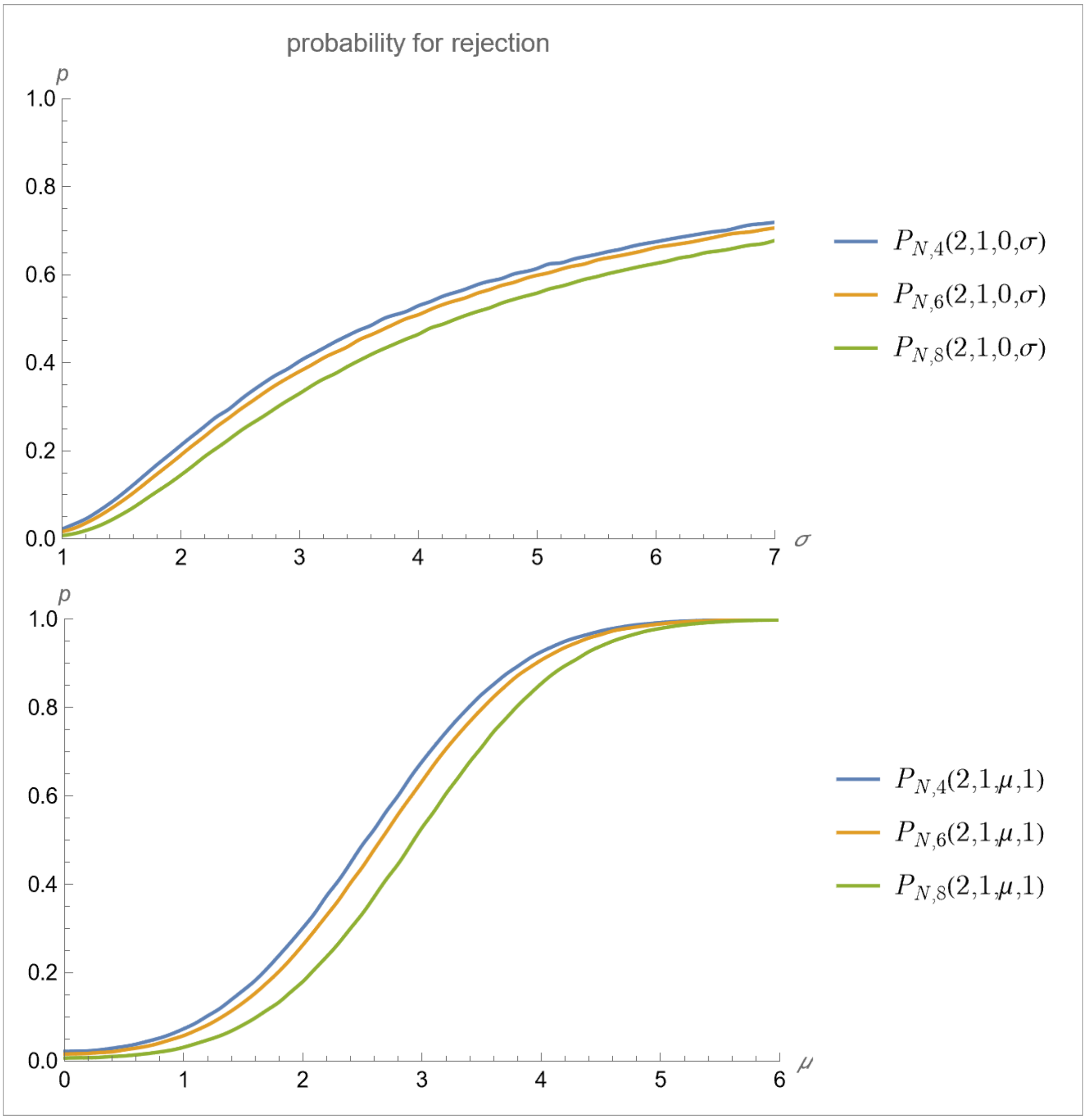

Fig 41: $P_{N,4}(2,1,\mu,\sigma)$ *vs* $P_{N,6}(2,1,\mu,\sigma)$ *vs* $P_{N,8}(2,1,\mu,\sigma)$

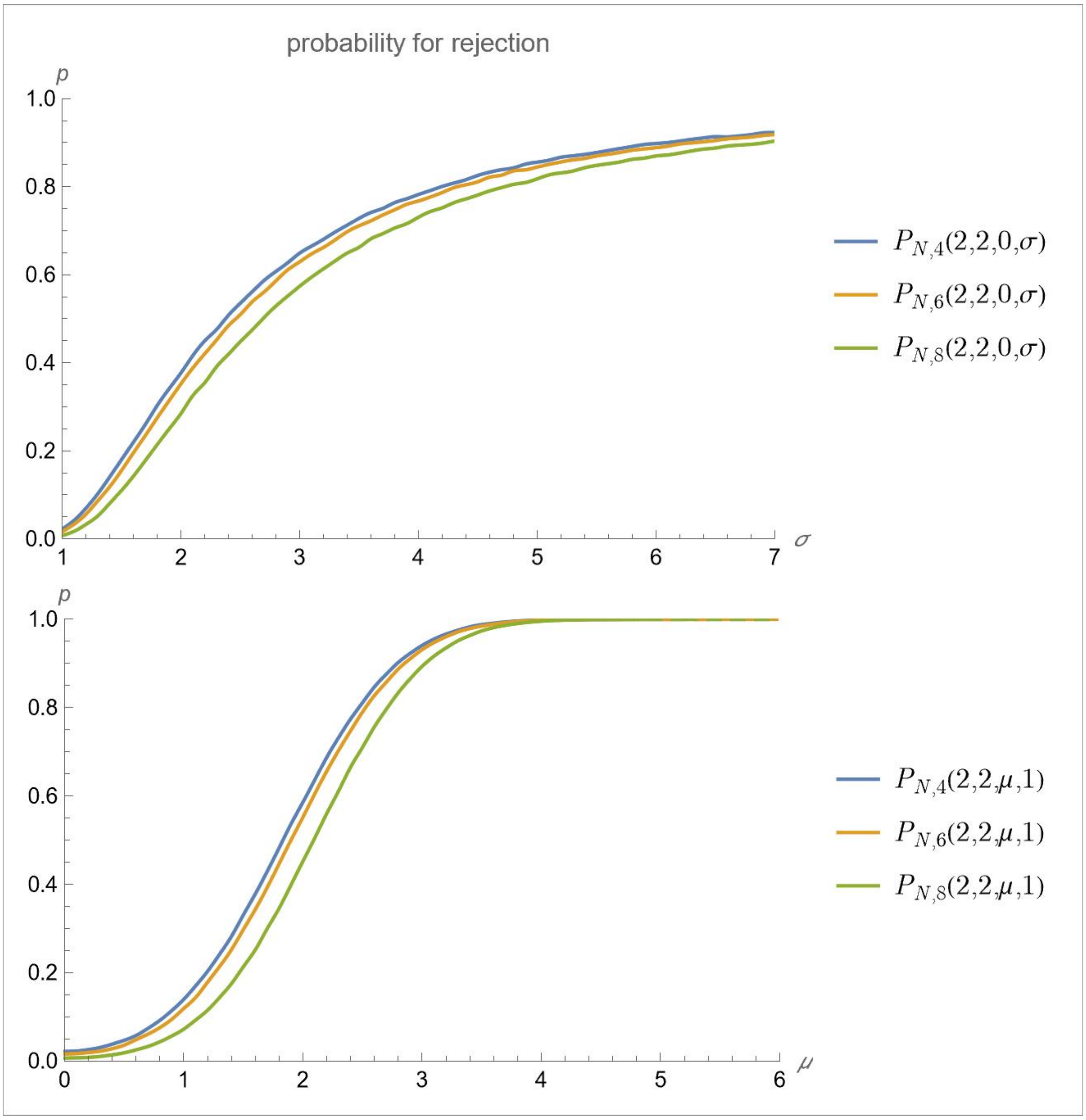

Fig 42: $P_{N,4}(2,2,\mu,\sigma)$ *vs* $P_{N,6}(2,2,\mu,\sigma)$ *vs* $P_{N,8}(2,2,\mu,\sigma)$

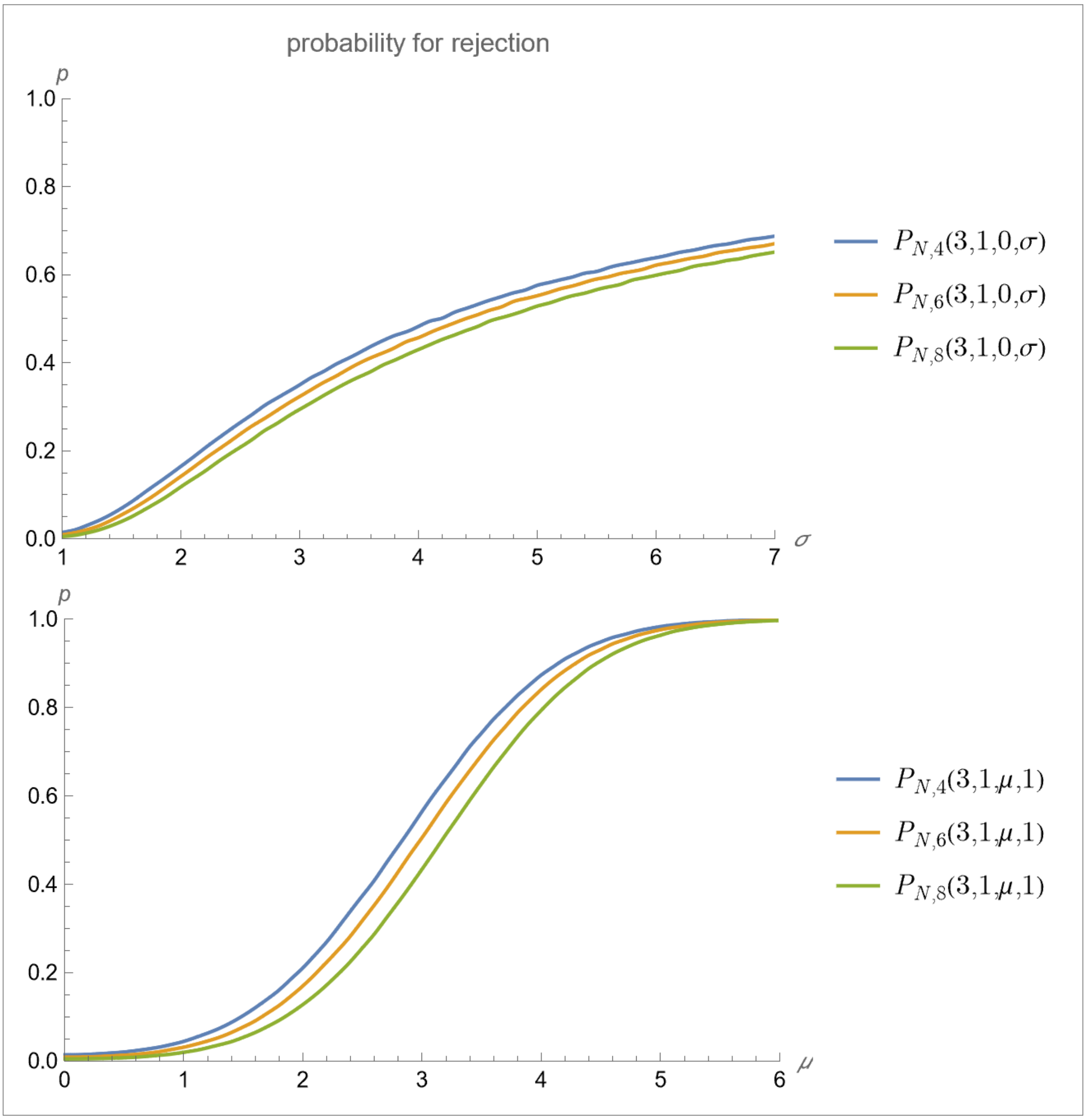

Fig 43: $P_{N,4}(3,1,\mu,\sigma)$ *vs* $P_{N,6}(3,1,\mu,\sigma)$ *vs* $P_{N,8}(3,1,\mu,\sigma)$

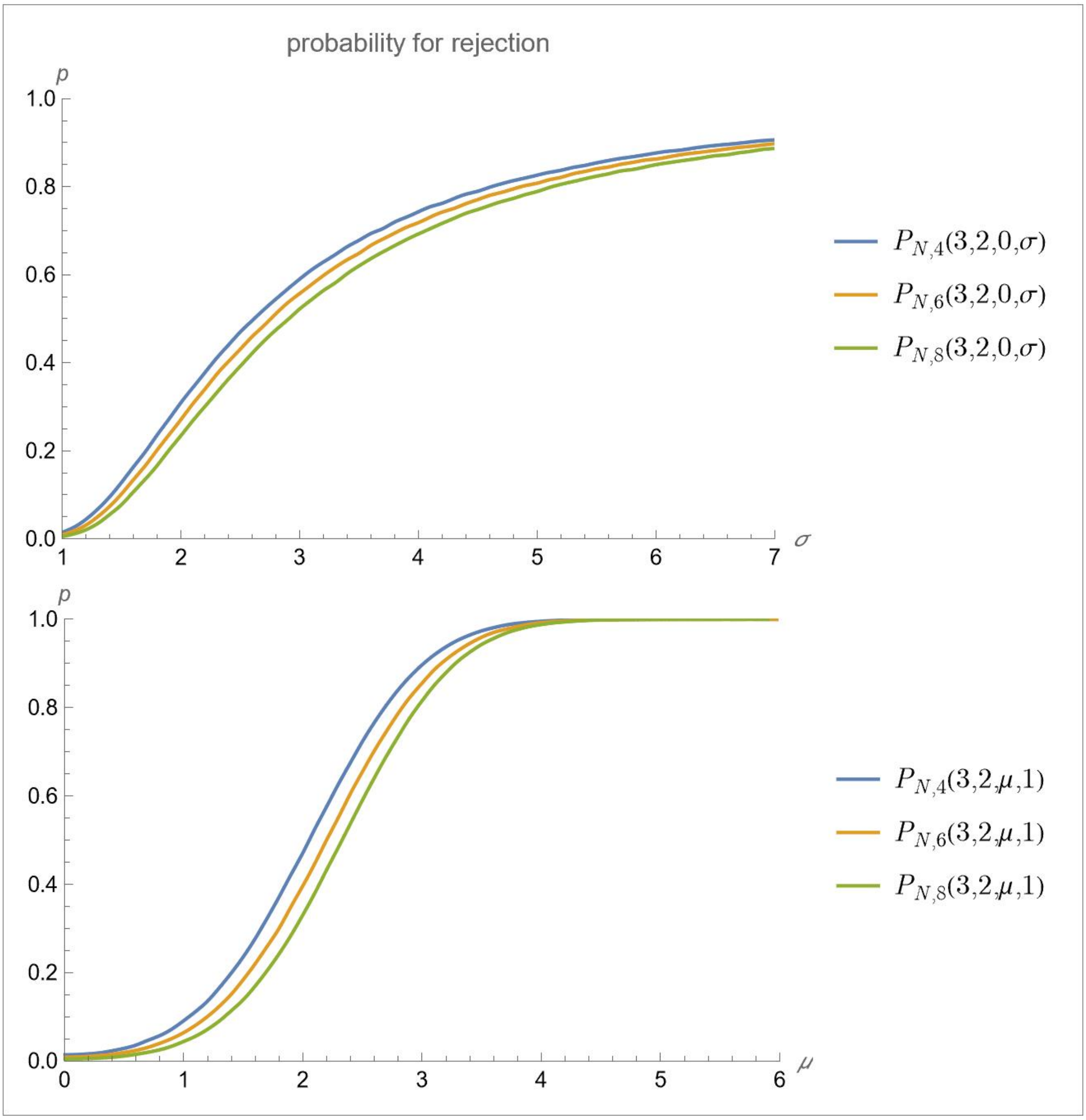

Fig 44: $P_{N,4}(3,2,\mu,\sigma)$ *vs* $P_{N,6}(3,2,\mu,\sigma)$ *vs* $P_{N,8}(3,2,\mu,\sigma)$

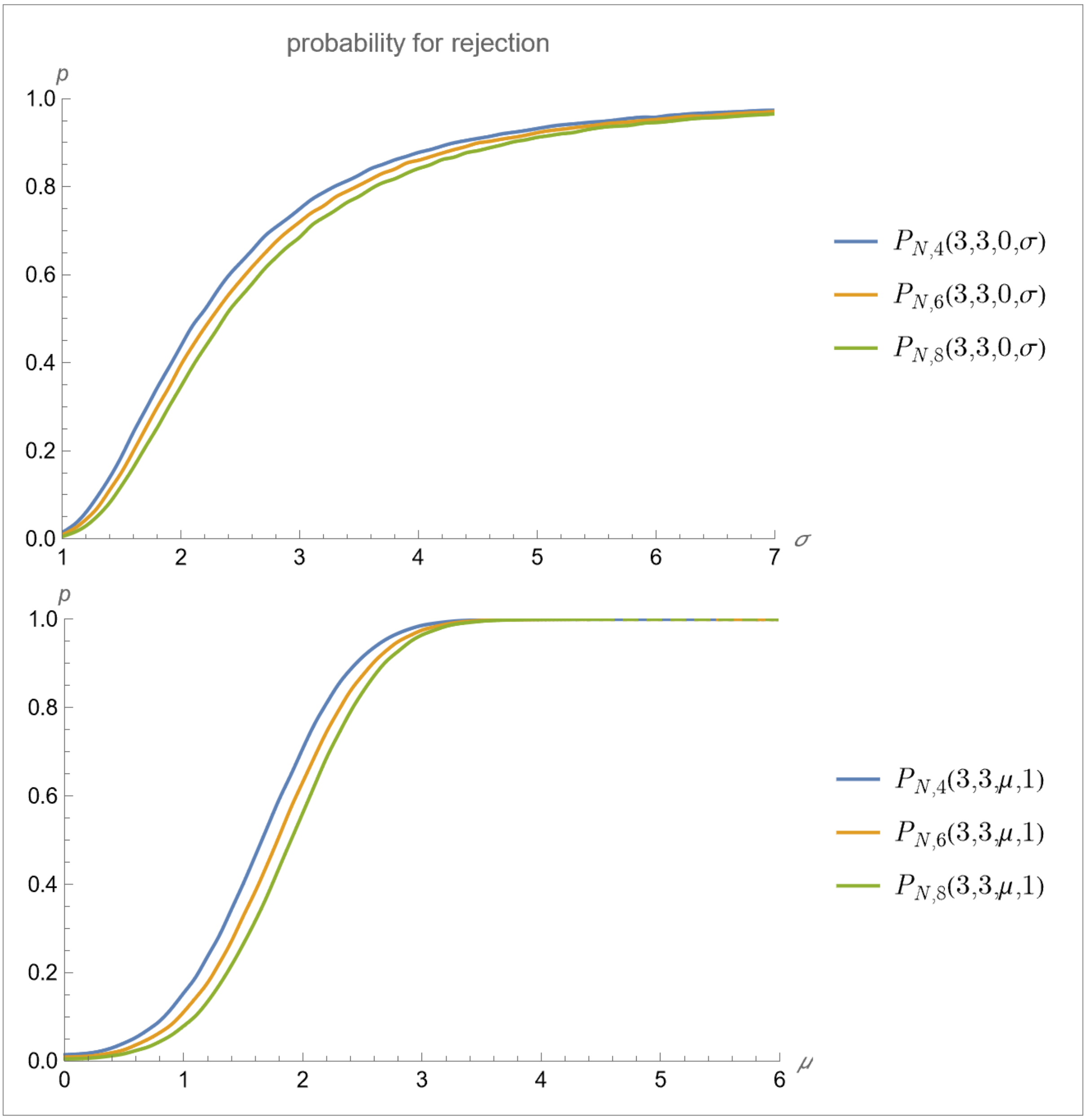

Fig 45: $P_{N,4}(3,3,\mu,\sigma)$ *vs* $P_{N,6}(3,3,\mu,\sigma)$ *vs* $P_{N,8}(3,3,\mu,\sigma)$

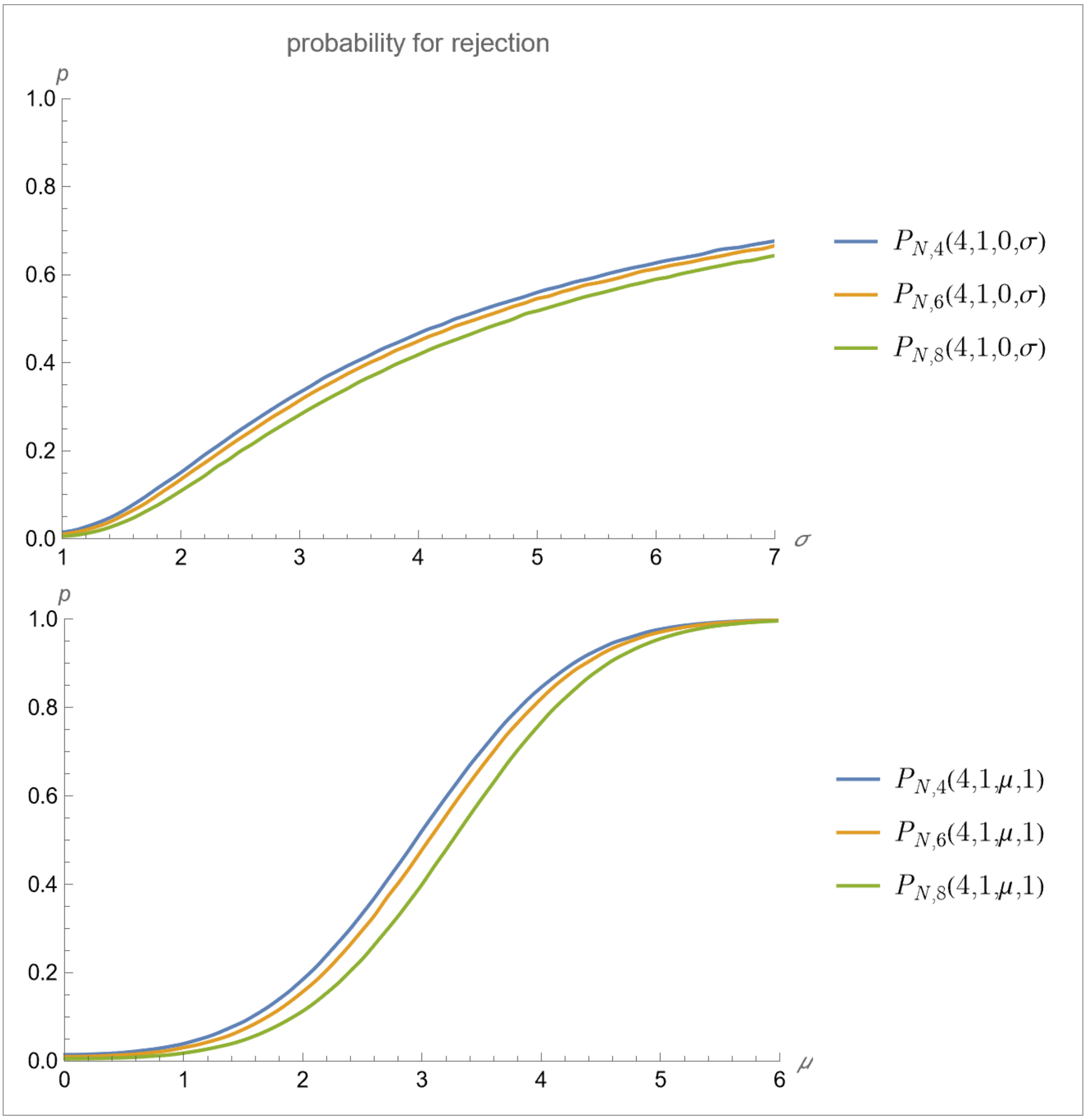

Fig 46: $P_{N,4}(4,1,\mu,\sigma)$ *vs* $P_{N,6}(4,1,\mu,\sigma)$ *vs* $P_{N,8}(4,1,\mu,\sigma)$

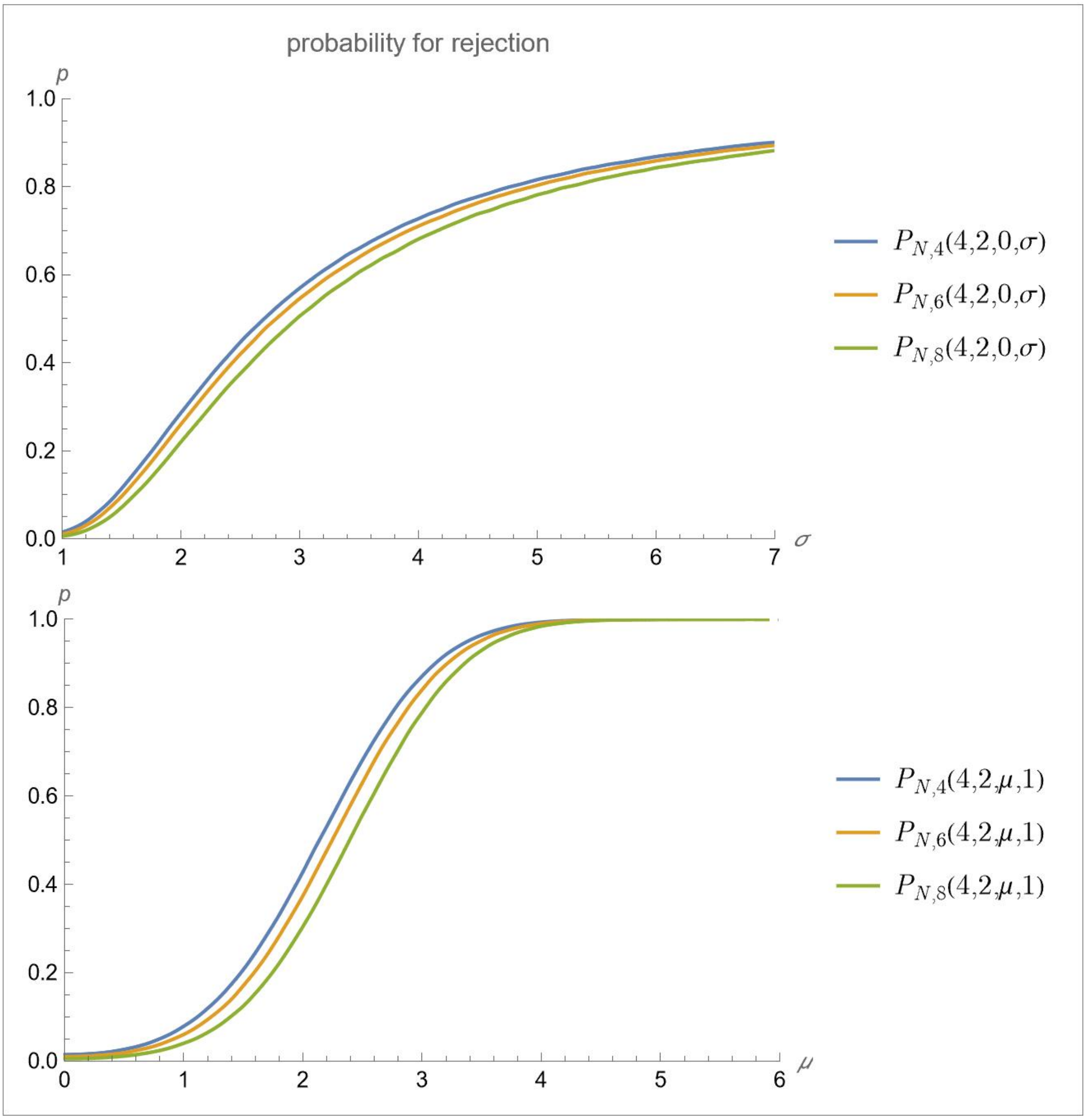

Fig 47: $P_{N,4}(4,2,\mu,\sigma)$ *vs* $P_{N,6}(4,2,\mu,\sigma)$ *vs* $P_{N,8}(4,2,\mu,\sigma)$

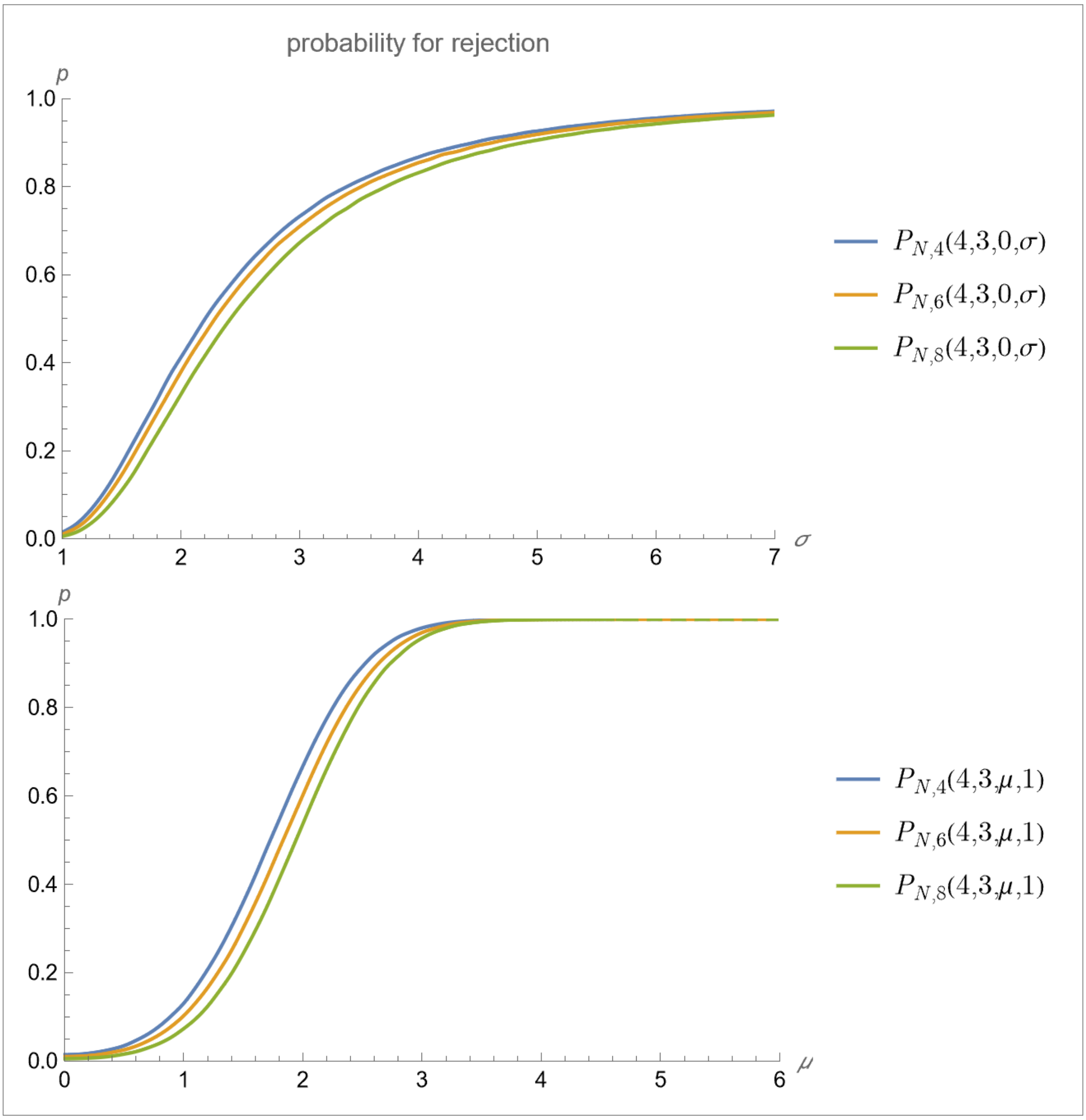

Fig 48: $P_{N,4}(4,3,\mu,\sigma)$ *vs* $P_{N,6}(4,3,\mu,\sigma)$ *vs* $P_{N,8}(4,3,\mu,\sigma)$

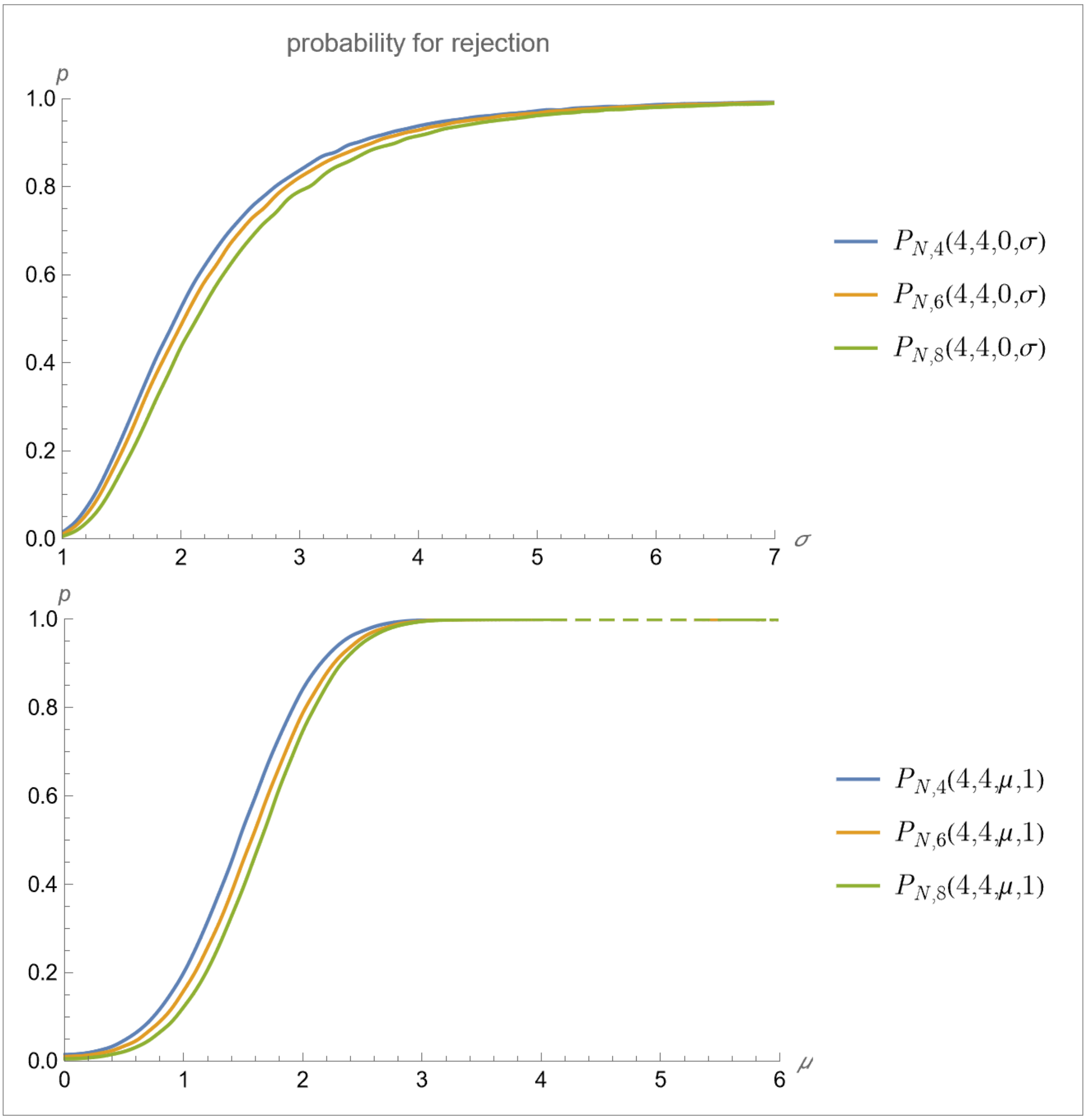

Fig 49: $P_{N,4}(4{,}4,\mu,\sigma)$ *vs* $P_{N,6}(4{,}4,\mu,\sigma)$ *vs* $P_{N,8}(4{,}4,\mu,\sigma)$

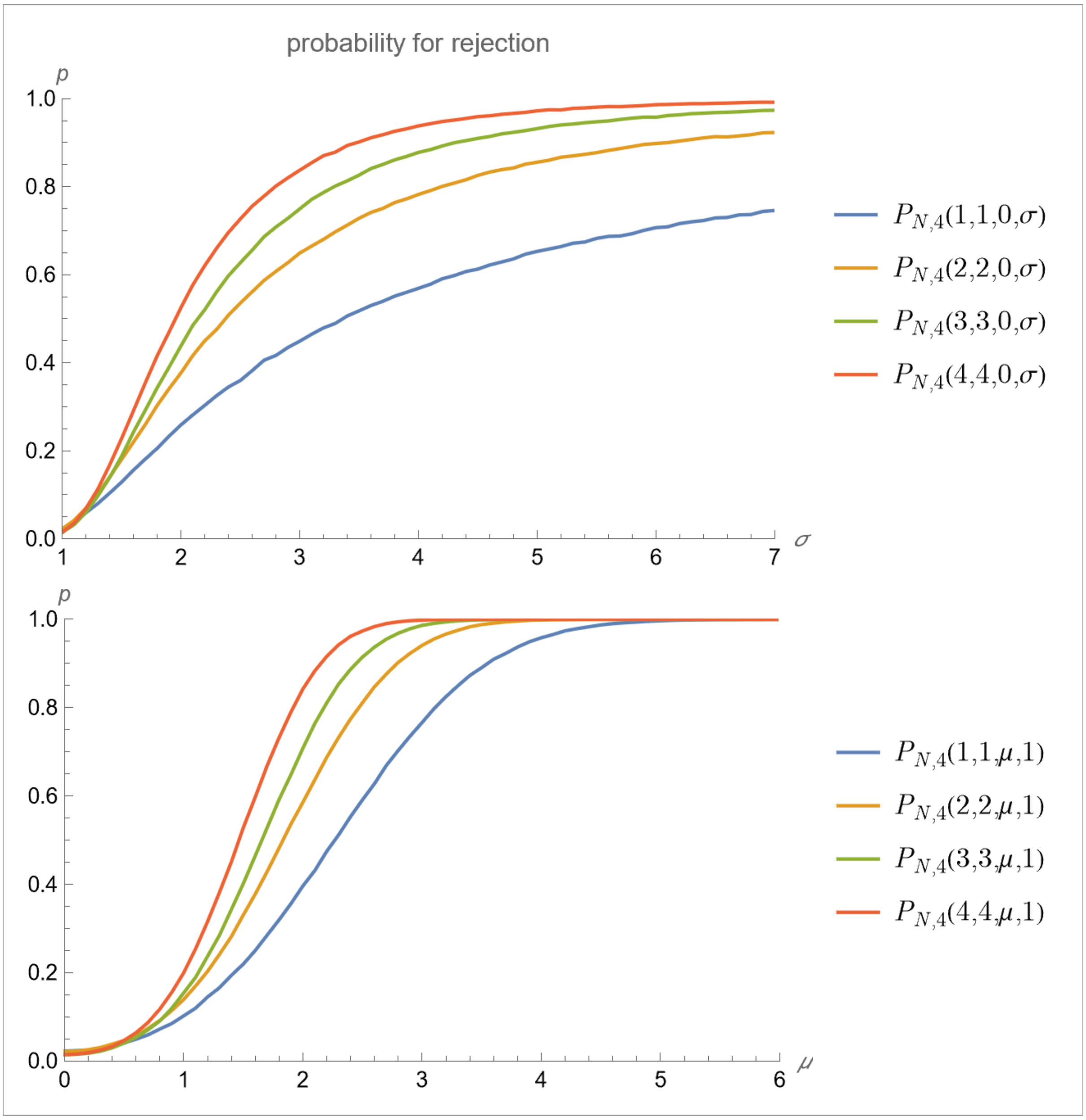

Fig 50: $P_{N,4}(1,1,\mu,\sigma)$ *vs* $P_{N,4}(2,2,\mu,\sigma)$ *vs* $P_{N,4}(3,3,\mu,\sigma)$ *vs* $P_{N,4}(4,4,\mu,\sigma)$

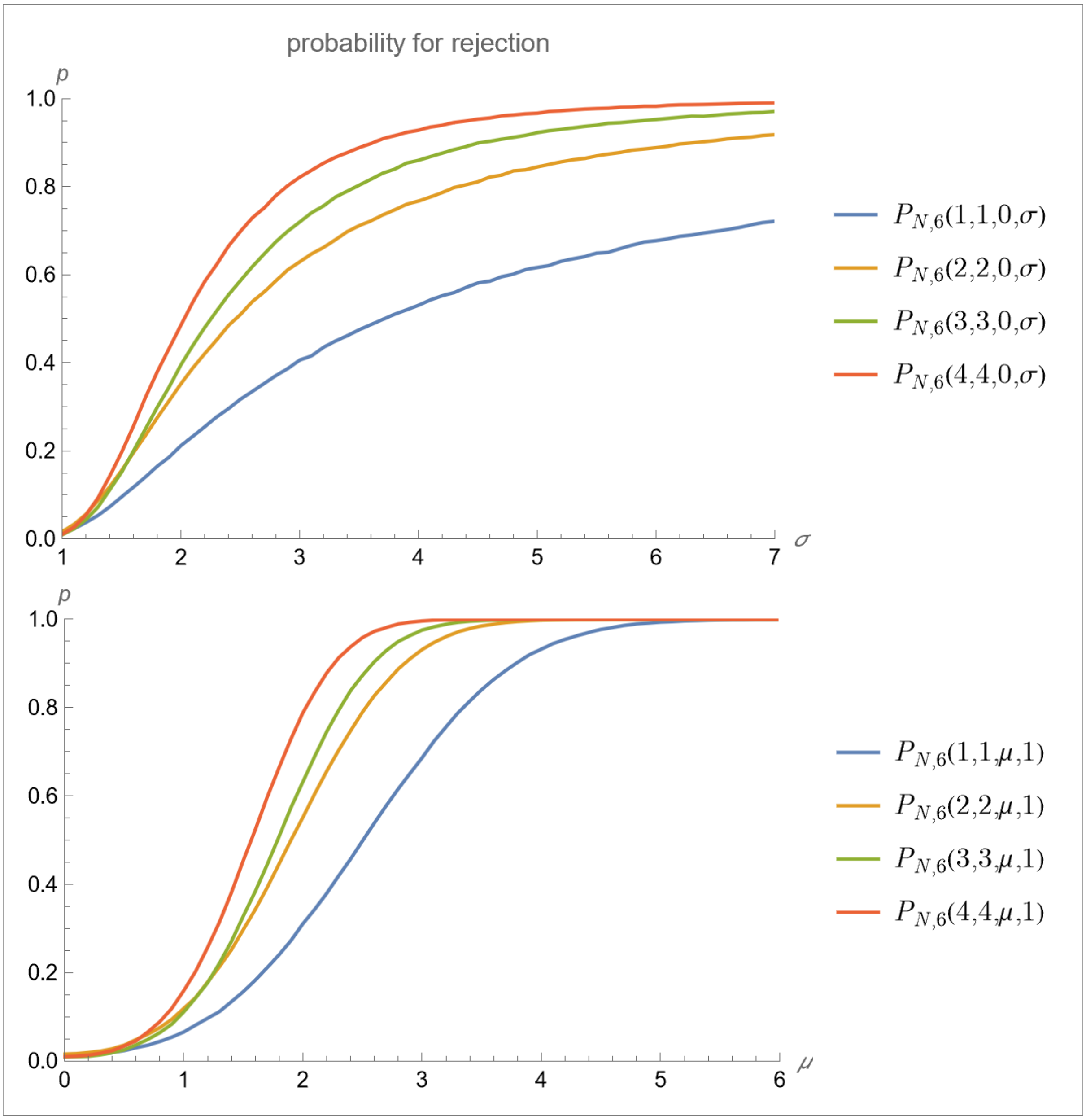

Fig 51: $P_{N,6}(1,1,\mu,\sigma)$ *vs* $P_{N,6}(2,2,\mu,\sigma)$ *vs* $P_{N,6}(3,3,\mu,\sigma)$ *vs* $P_{N,6}(4,4,\mu,\sigma)$

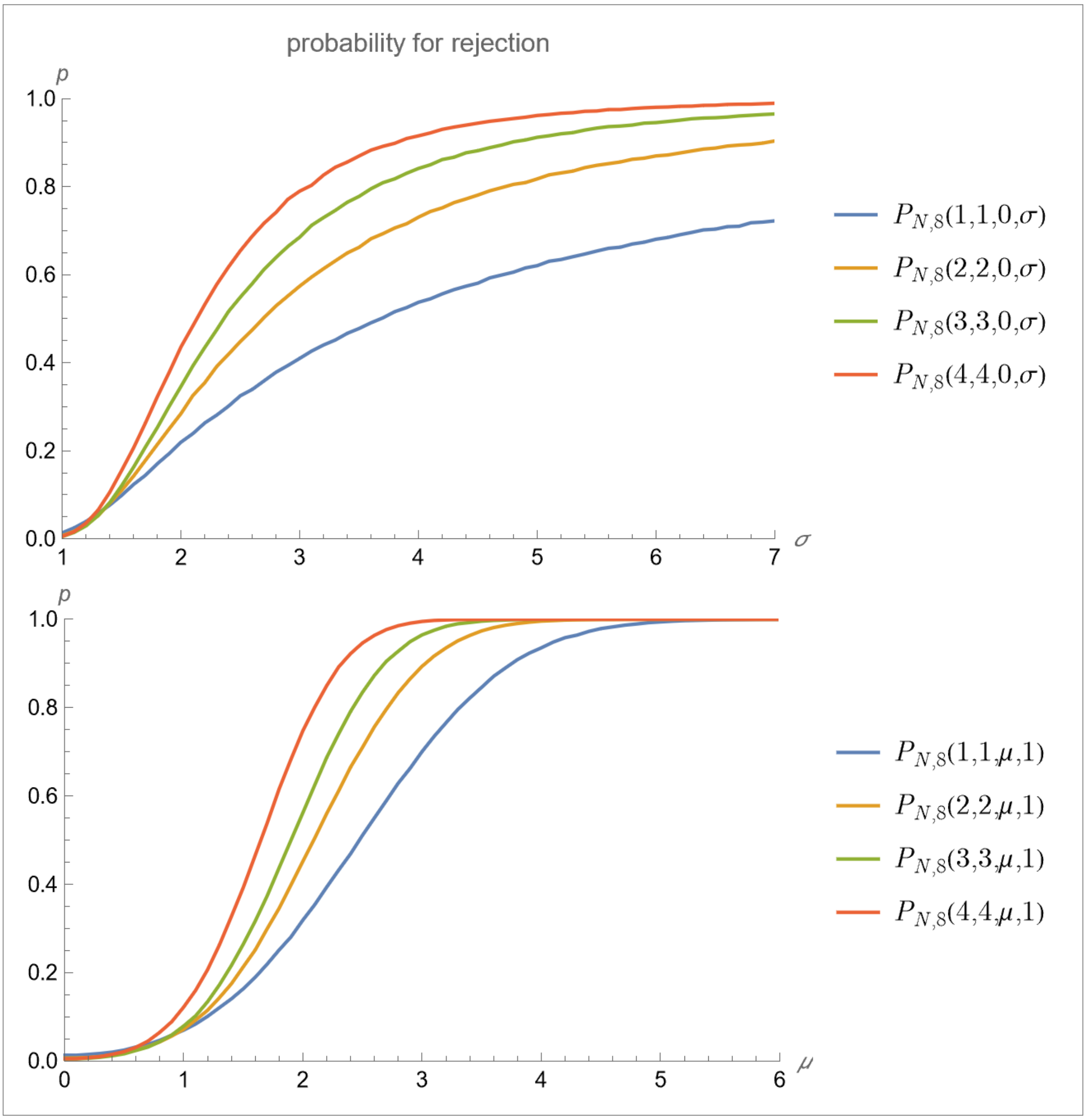

Fig 52: $P_{N,8}(1,1,\mu,\sigma)$ *vs* $P_{N,8}(2,2,\mu,\sigma)$ *vs* $P_{N,8}(3,3,\mu,\sigma)$ *vs* $P_{N,8}(4,4,\mu,\sigma)$

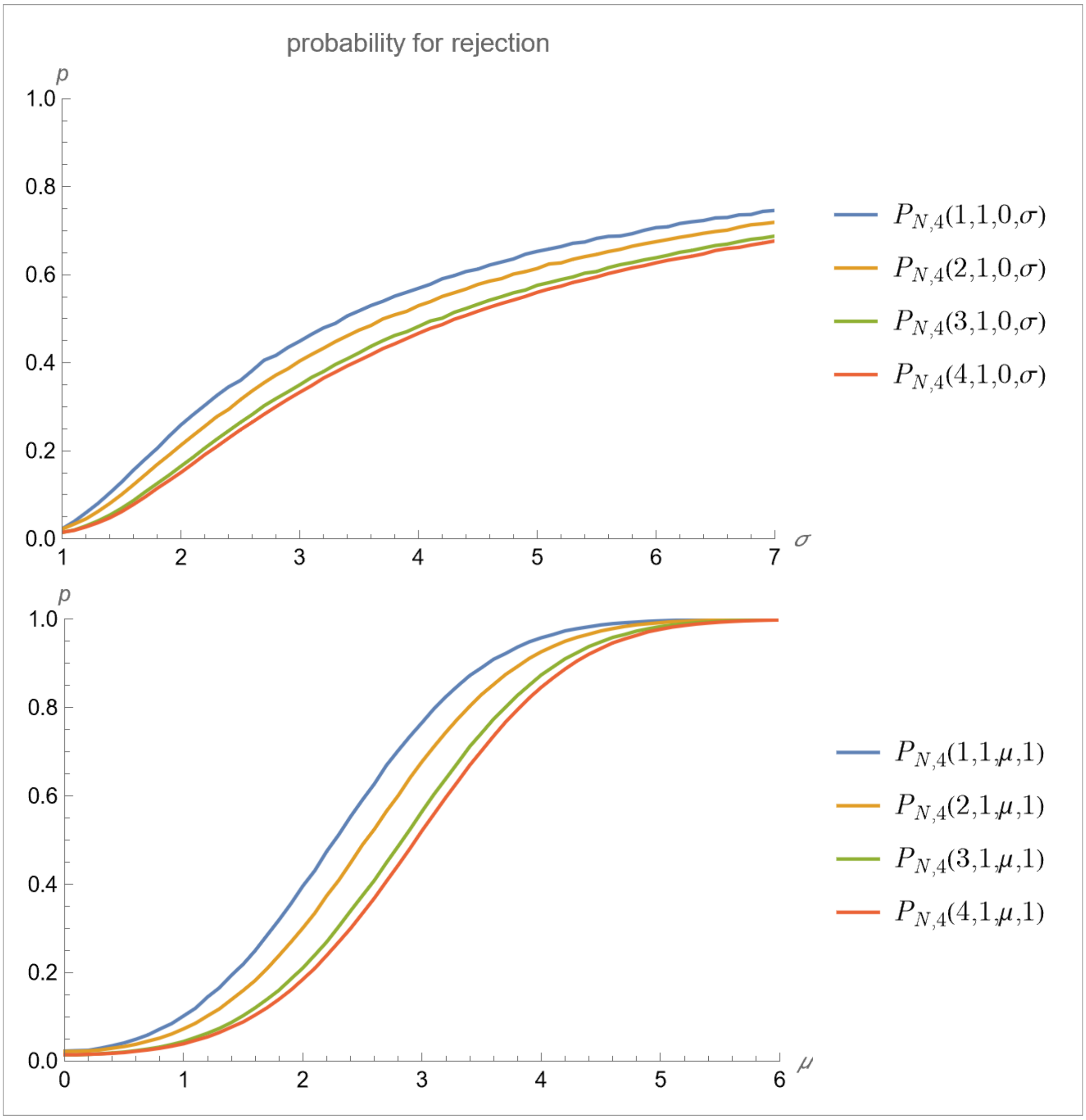

Fig 53: $P_{N,4}(1,1,\mu,\sigma)$ *vs* $P_{N,4}(2,1,\mu,\sigma)$ *vs* $P_{N,4}(3,1,\mu,\sigma)$ *vs* $P_{N,4}(4,1,\mu,\sigma)$

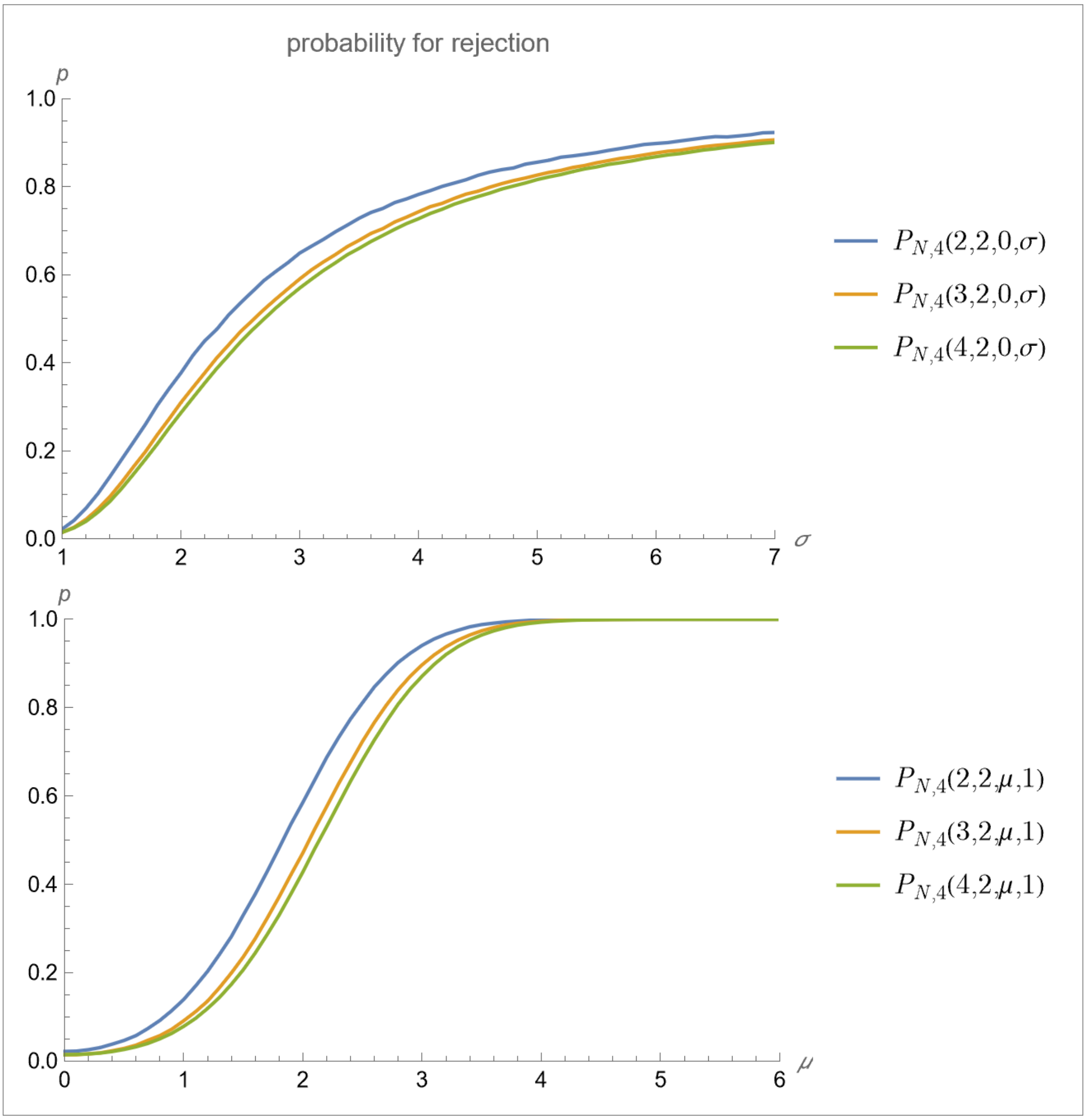

Fig 54: $P_{N,4}(2,2,\mu,\sigma)$ *vs* $P_{N,4}(3,2,\mu,\sigma)$ *vs* $P_{N,4}(4,2,\mu,\sigma)$

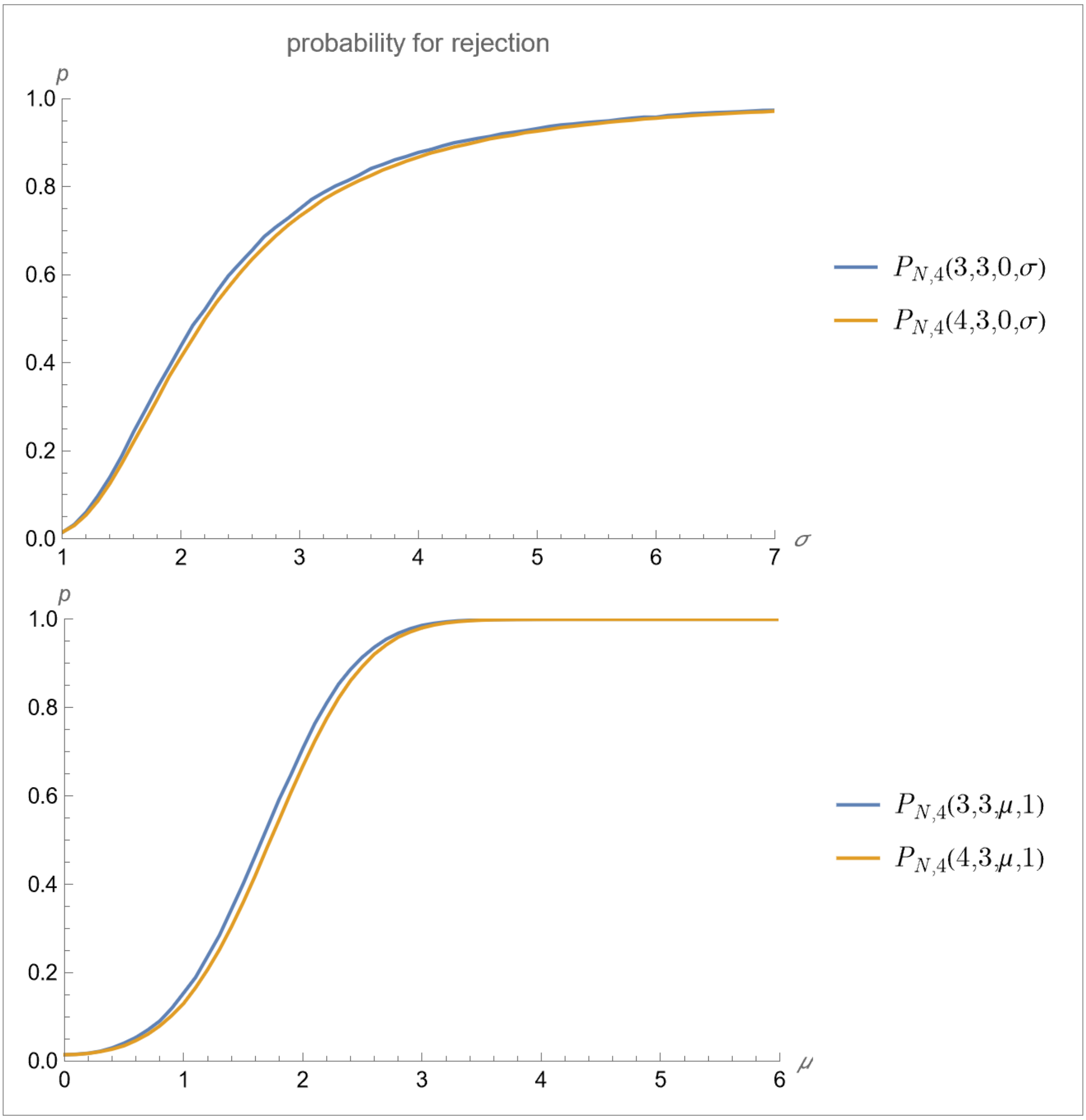

Fig 55: $P_{N,4}(3,3,\mu,\sigma)$ *vs* $P_{N,4}(4,3,\mu,\sigma)$

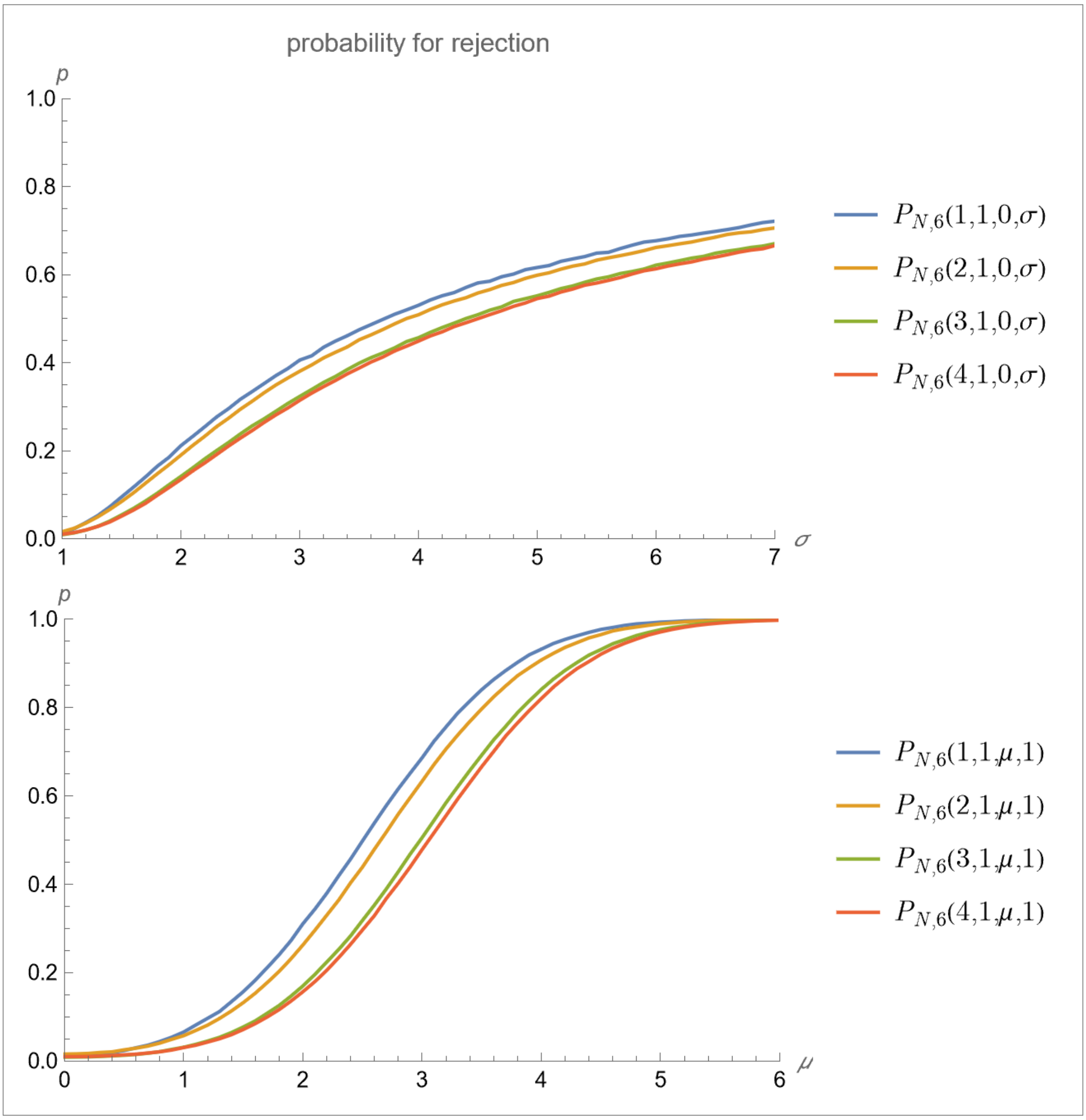

Fig 56: $P_{N,6}(1,1,\mu,\sigma)$ *vs* $P_{N,6}(2,1,\mu,\sigma)$ *vs* $P_{N,6}(3,1,\mu,\sigma)$ *vs* $P_{N,6}(4,1,\mu,\sigma)$

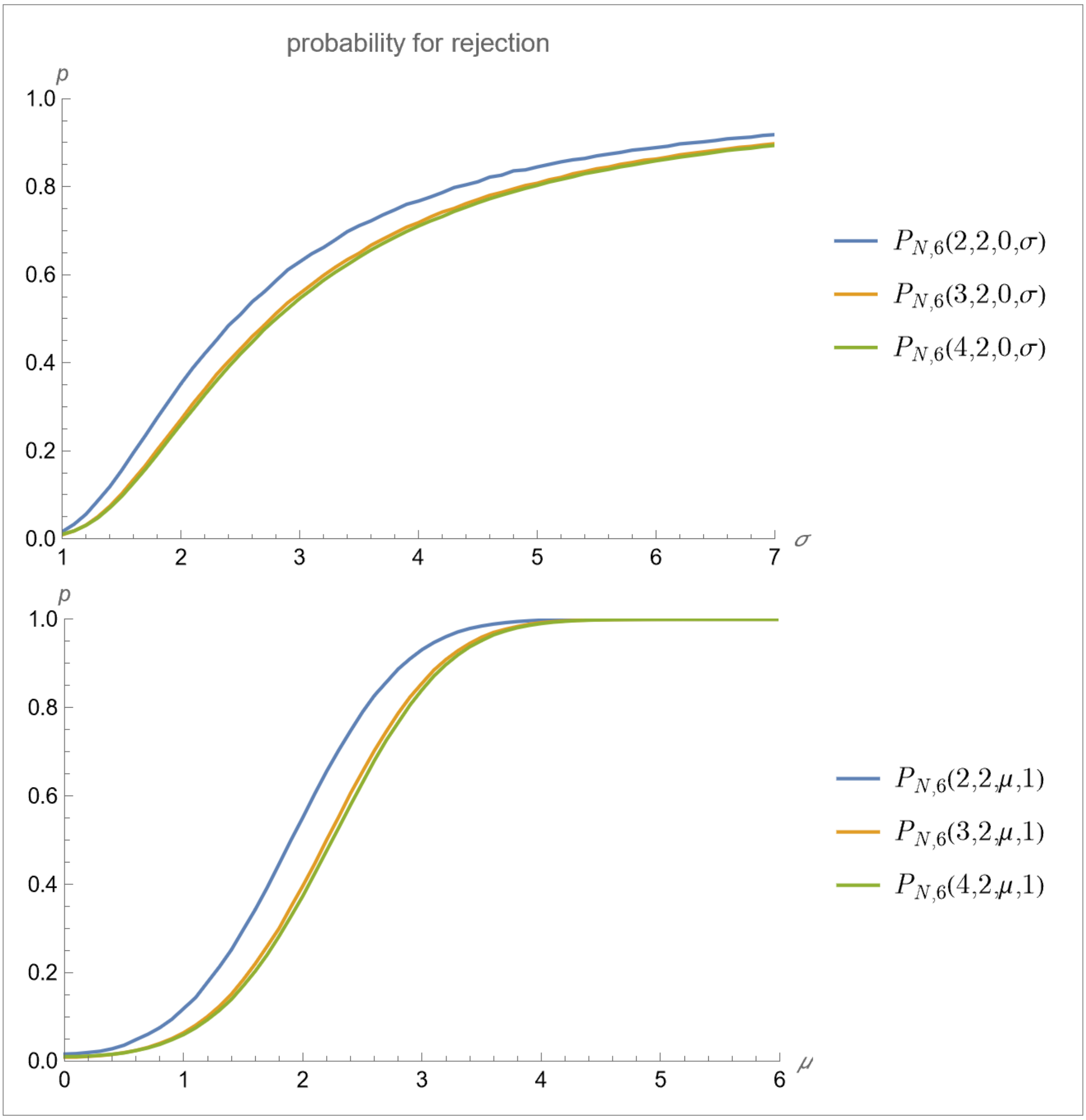

Fig 57: $P_{N,6}(2{,}2,\mu,\sigma)$ *vs* $P_{N,6}(3{,}2,\mu,\sigma)$ *vs* $P_{N,6}(4{,}2,\mu,\sigma)$

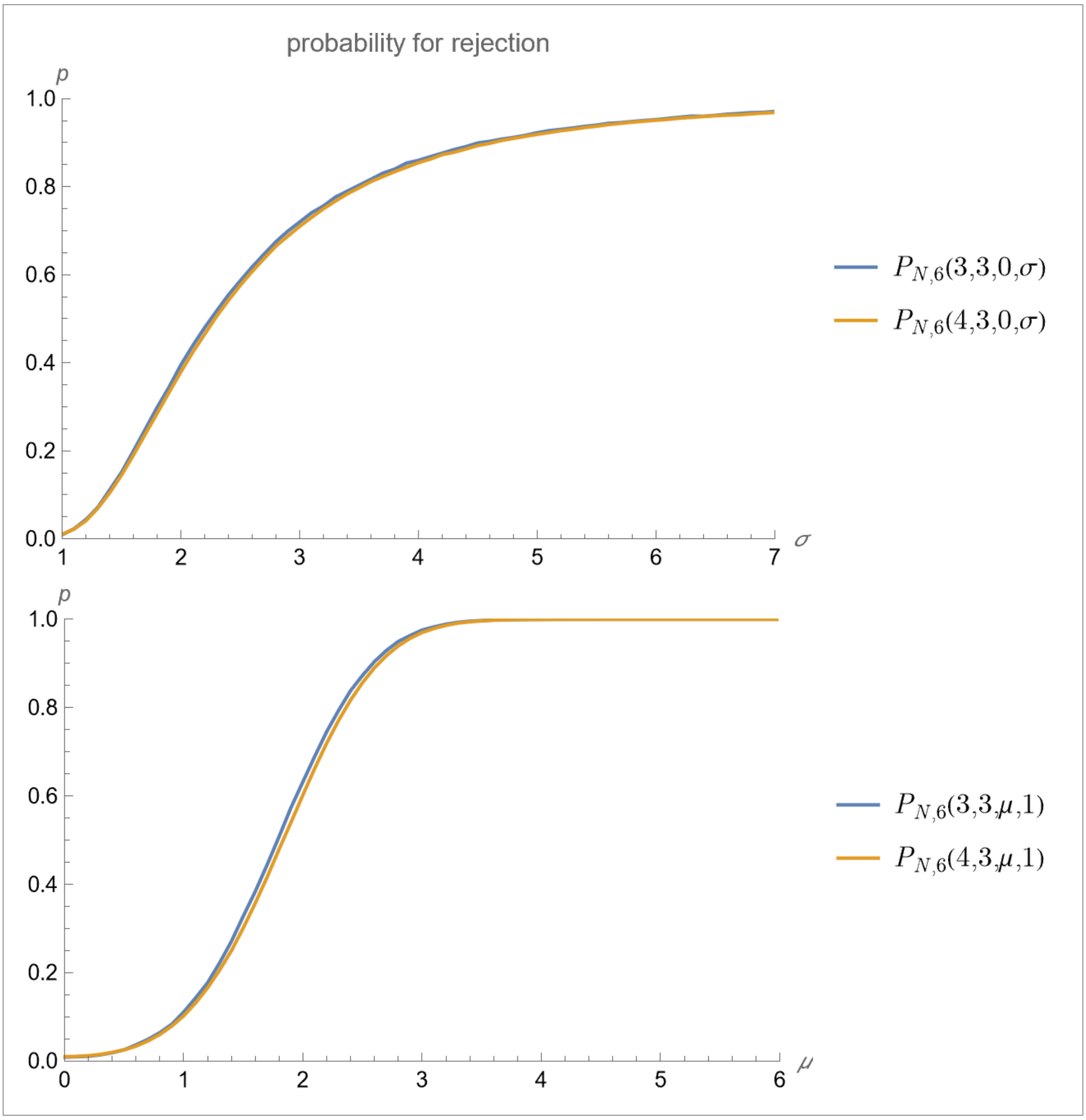

Fig 58: $P_{N,6}(3,3,\mu,\sigma)$ *vs* $P_{N,6}(4,3,\mu,\sigma)$

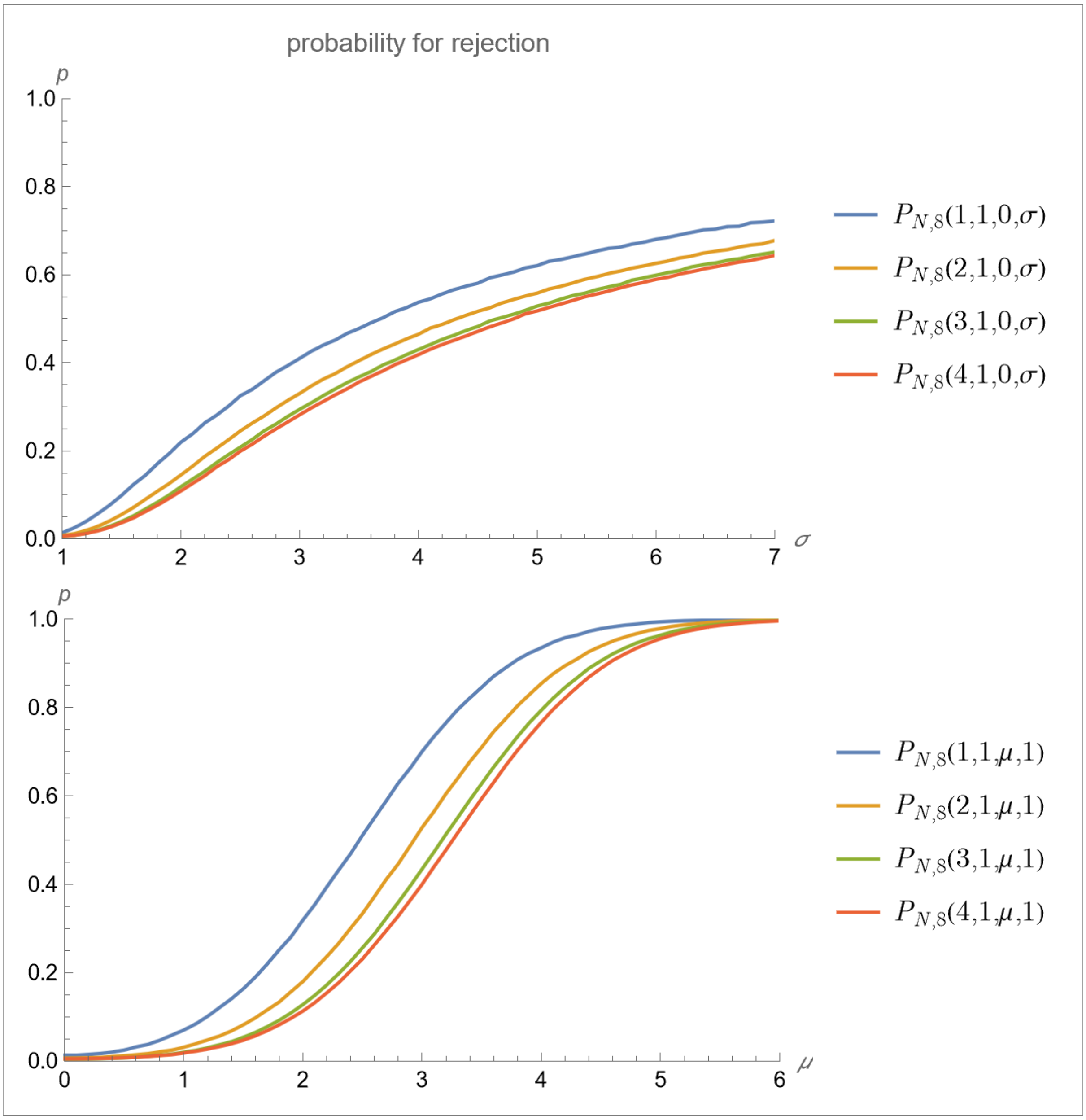

Fig 59: $P_{N,8}(1,1,\mu,\sigma)$ *vs* $P_{N,8}(2,1,\mu,\sigma)$ *vs* $P_{N,8}(3,1,\mu,\sigma)$ *vs* $P_{N,8}(4,1,\mu,\sigma)$

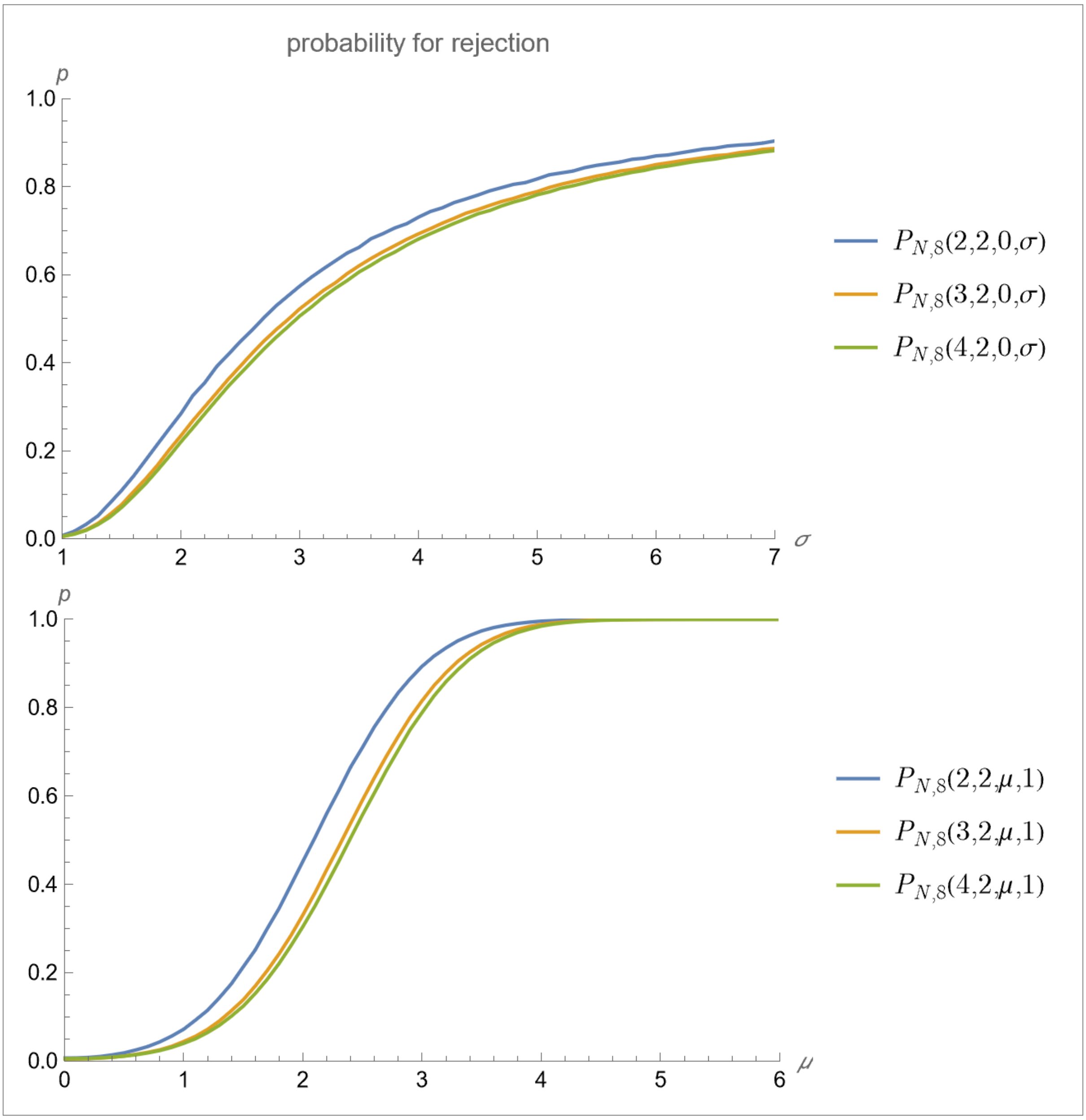

Fig 60: $P_{N,8}(2,2,\mu,\sigma)$ *vs* $P_{N,8}(3,2,\mu,\sigma)$ *vs* $P_{N,8}(4,2,\mu,\sigma)$

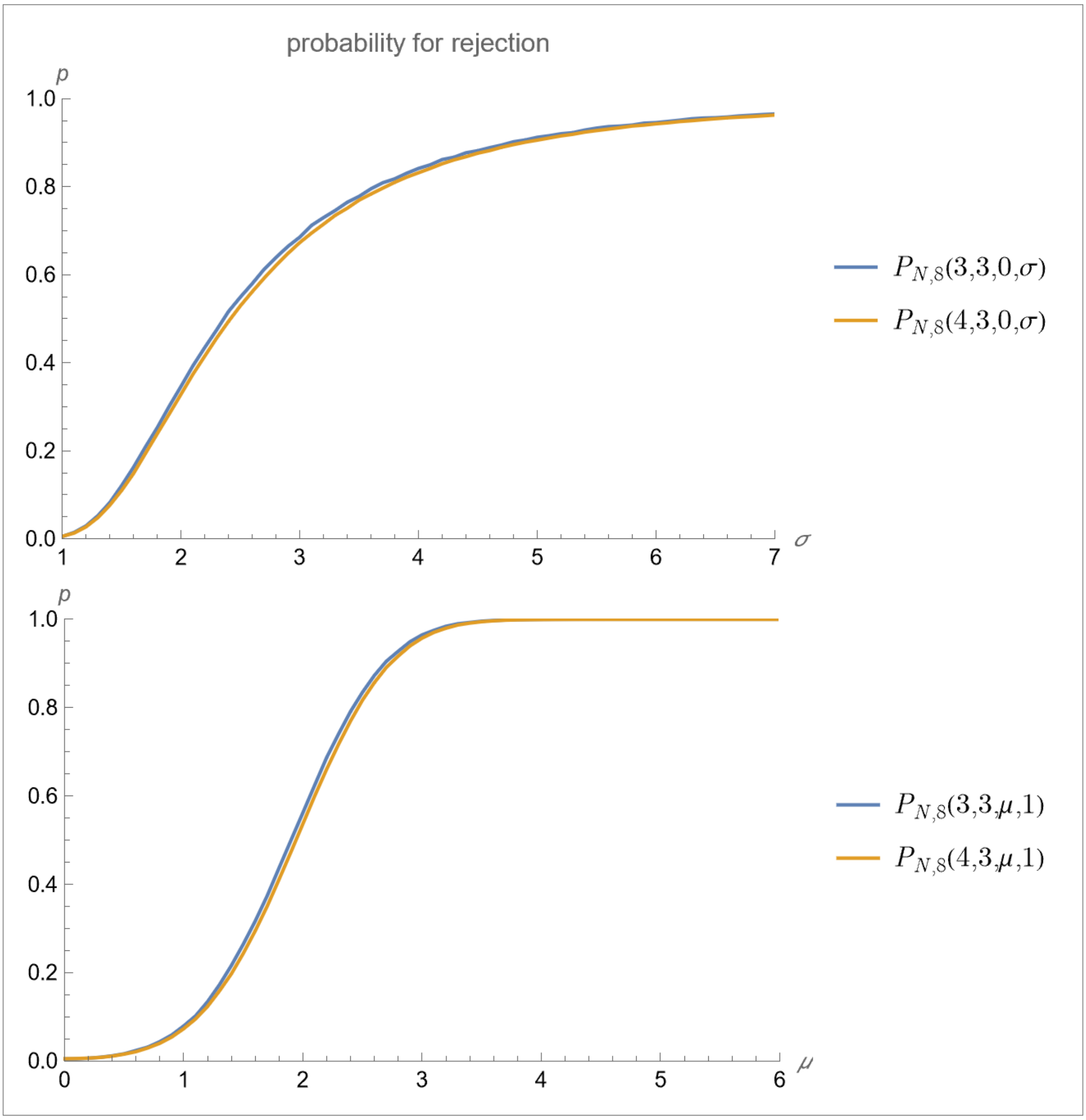

Fig 61: $P_{N,6}(3,3,\mu,\sigma)$ *vs* $P_{N,6}(4,3,\mu,\sigma)$

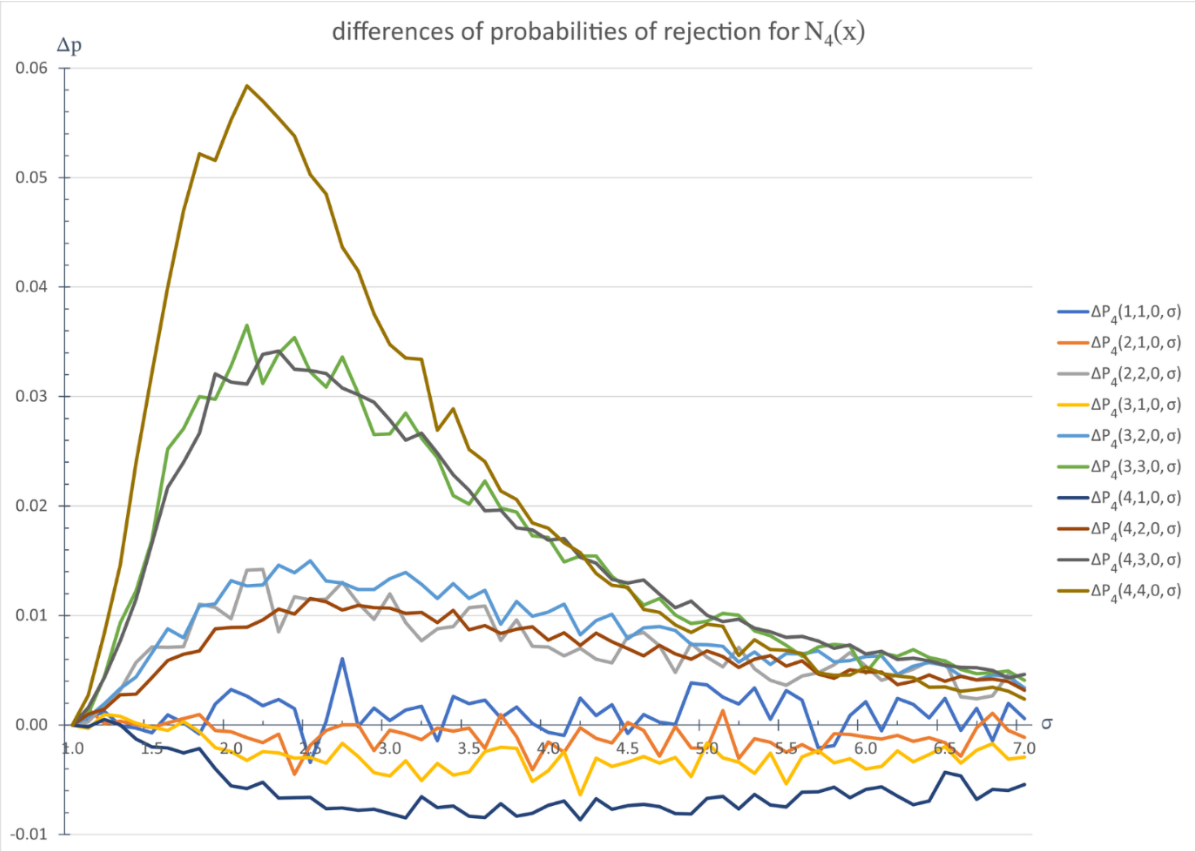

Fig 62: $\Delta P_4(n, k, 0, \sigma)$

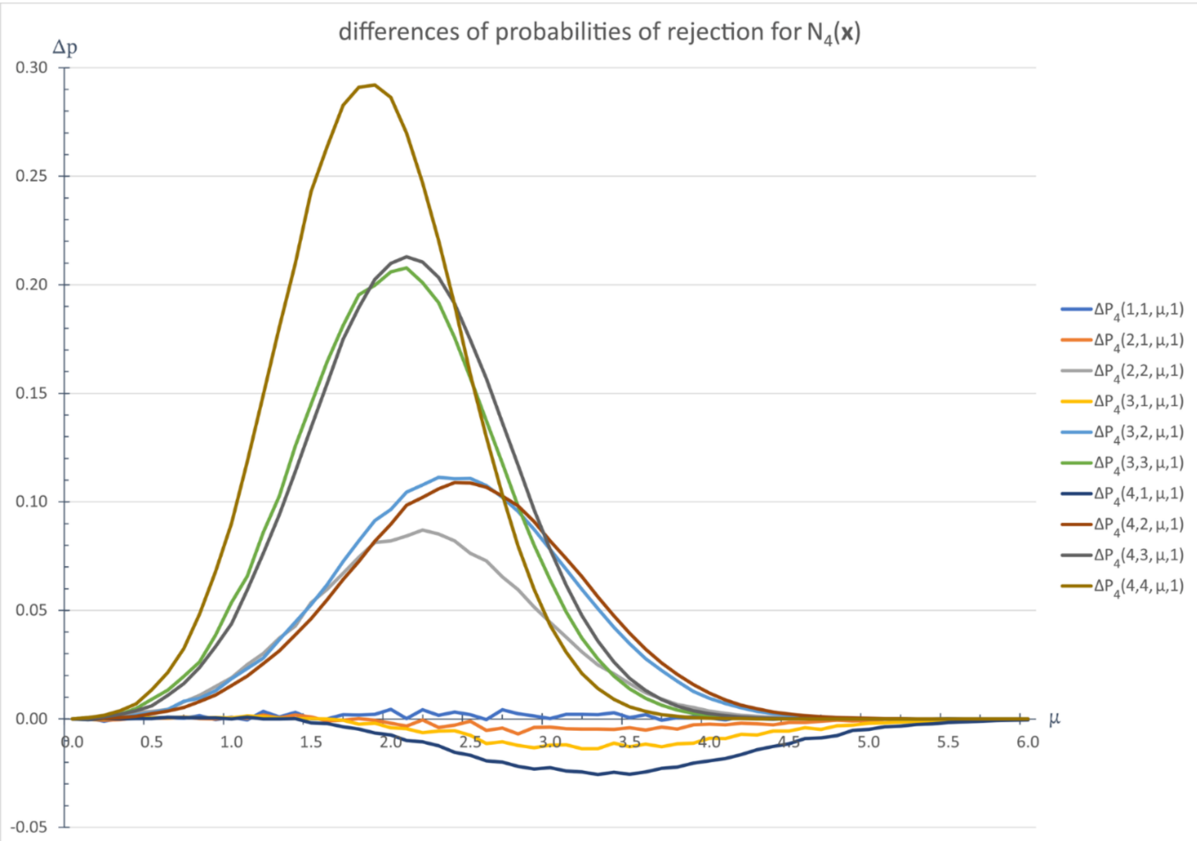

Fig 63: $\Delta P_4(n, k, \mu, 1)$

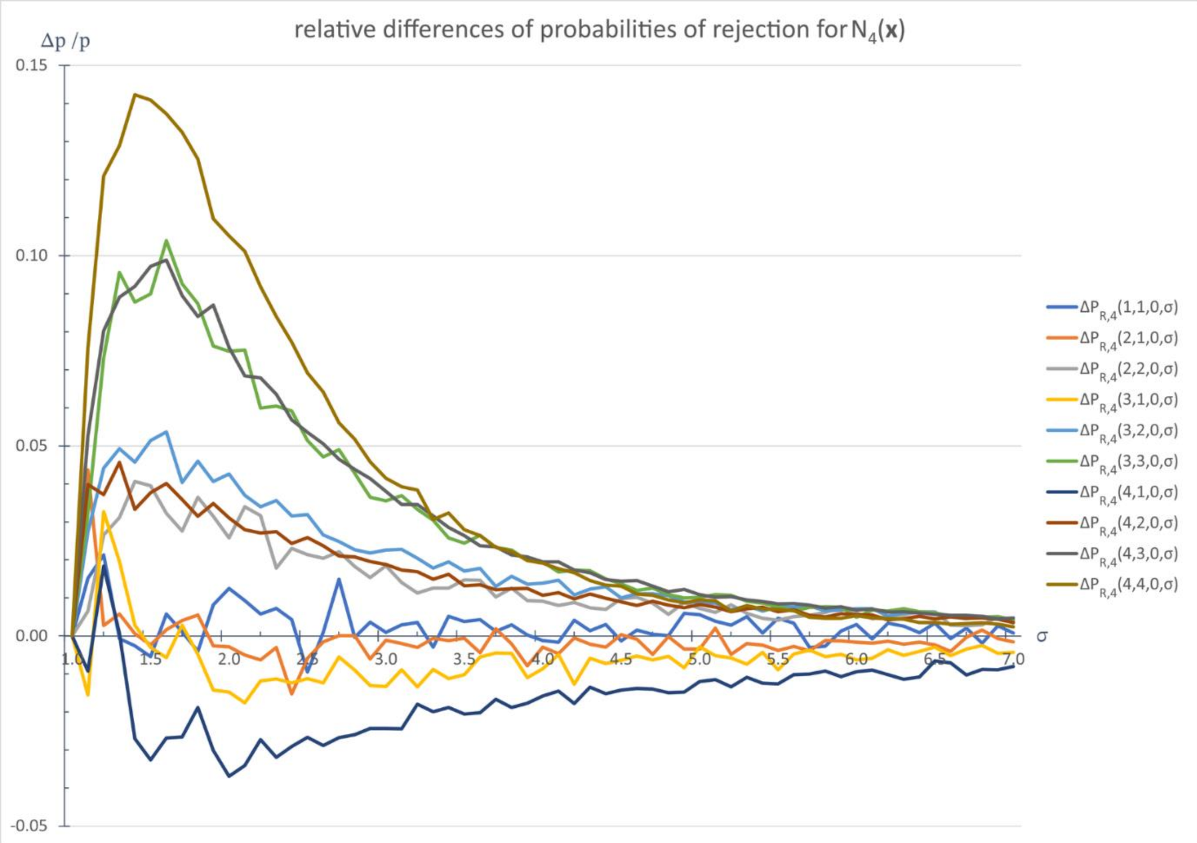

Fig 64: $\Delta P_{R,4}(n, k, 0, \sigma)$

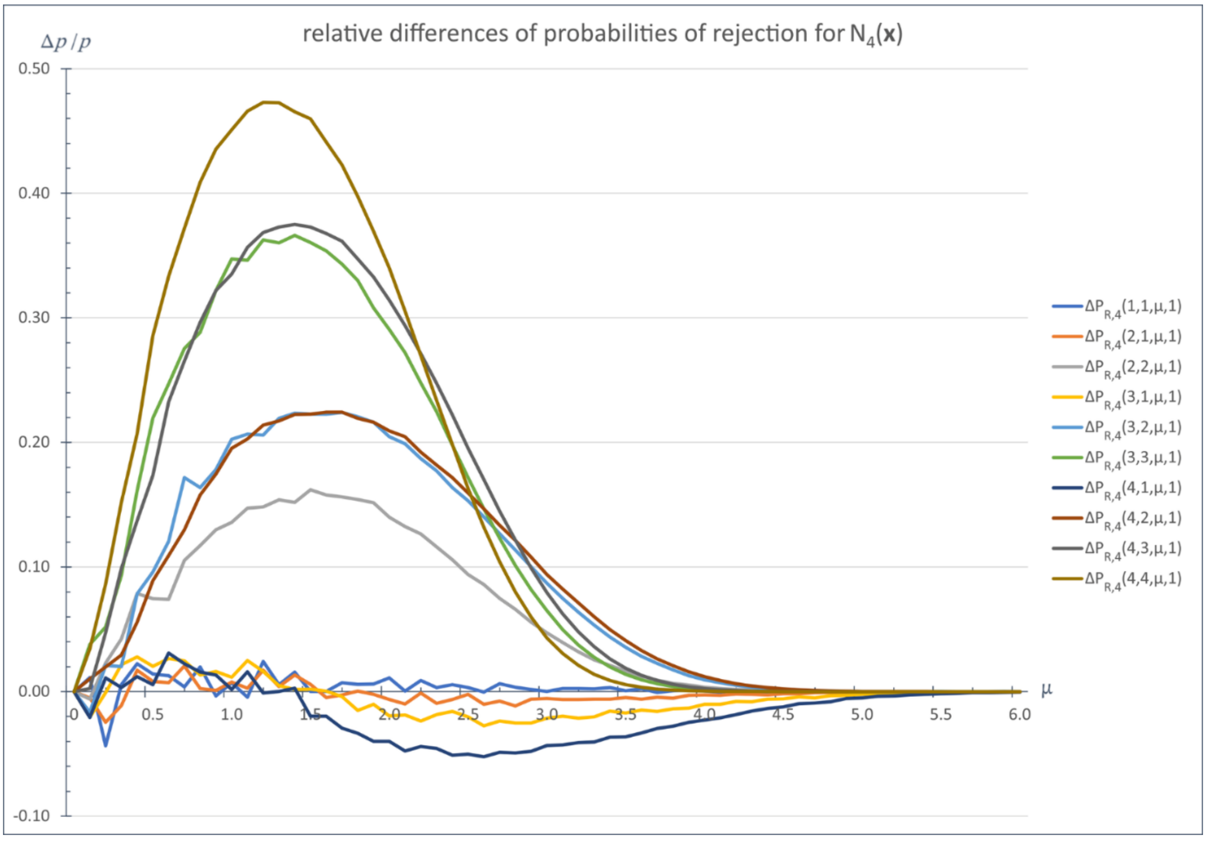

Fig 65: $\Delta P_{R,4}(n, k, \mu, 1)$

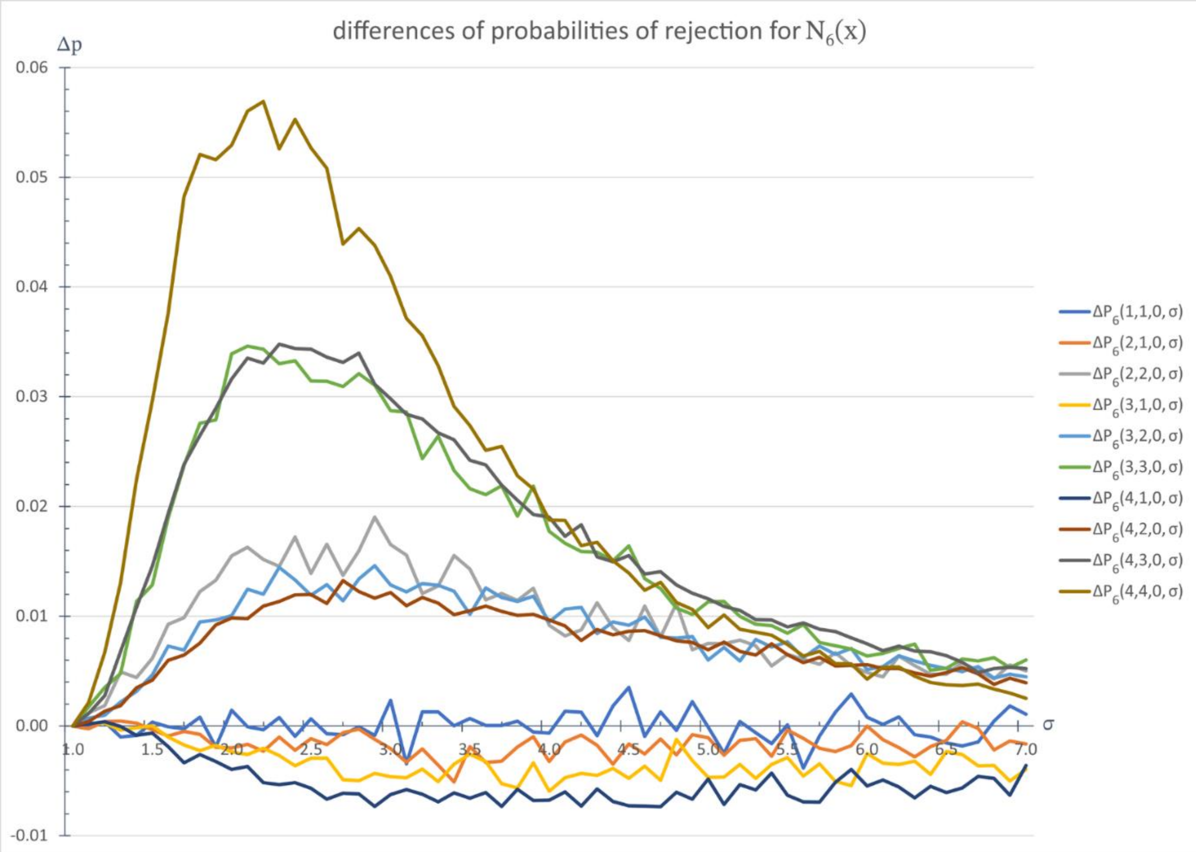

Fig 66: $\Delta P_6(n,k,0,\sigma)$

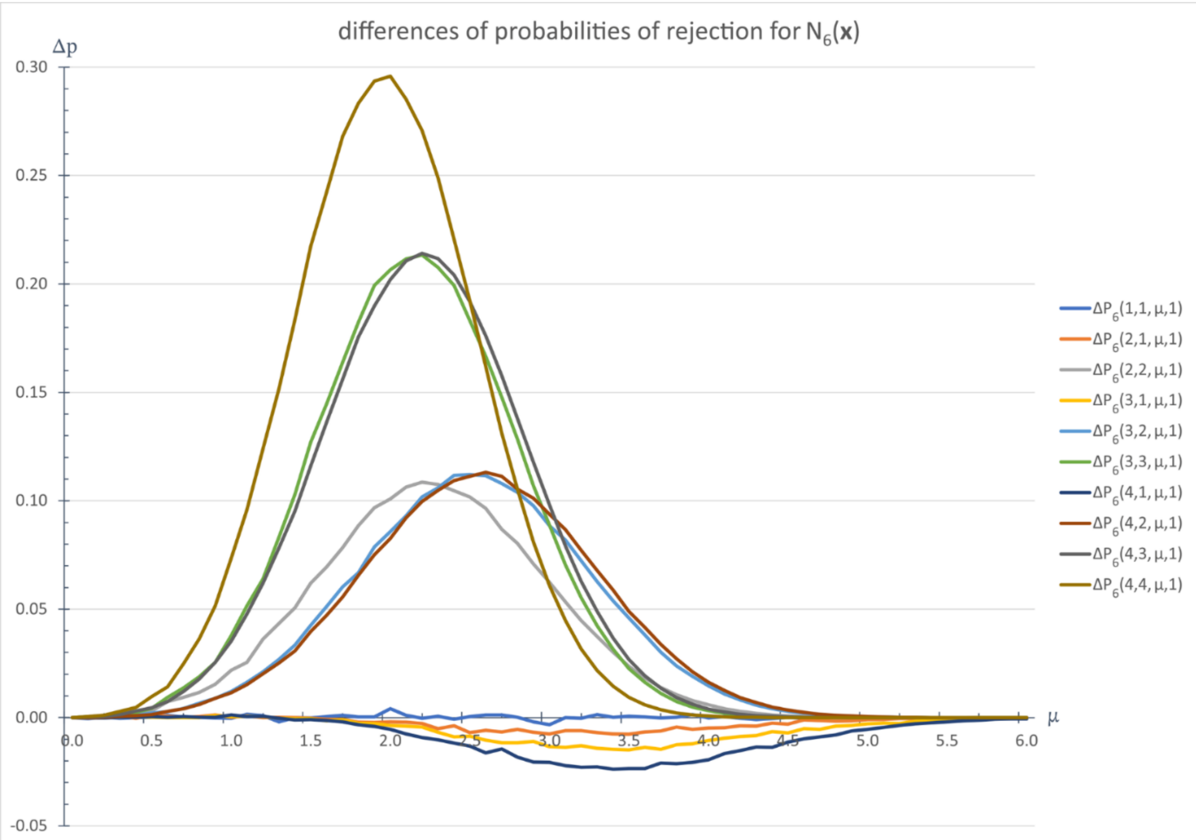

Fig 67: $\Delta P_6(n,k,\mu,1)$

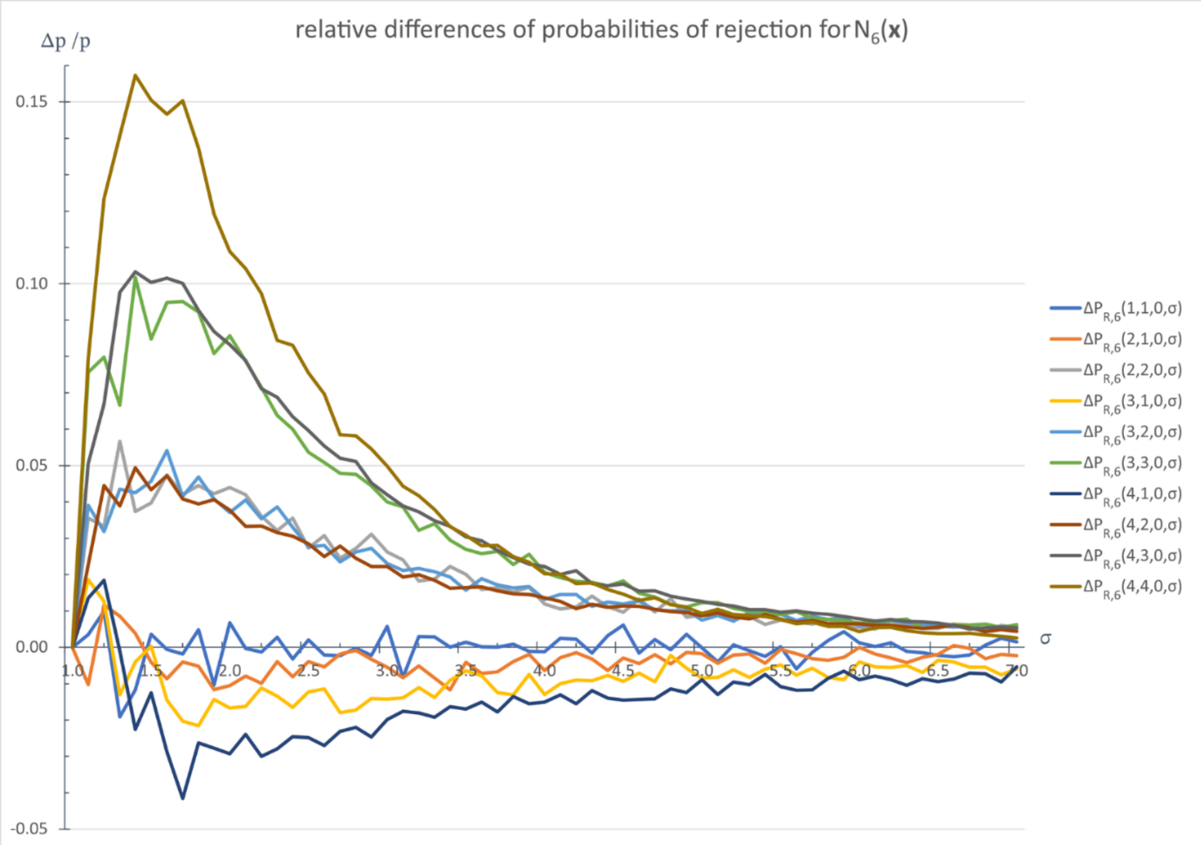

Fig 68: $\Delta P_{R,6}(n, k, 0, \sigma)$

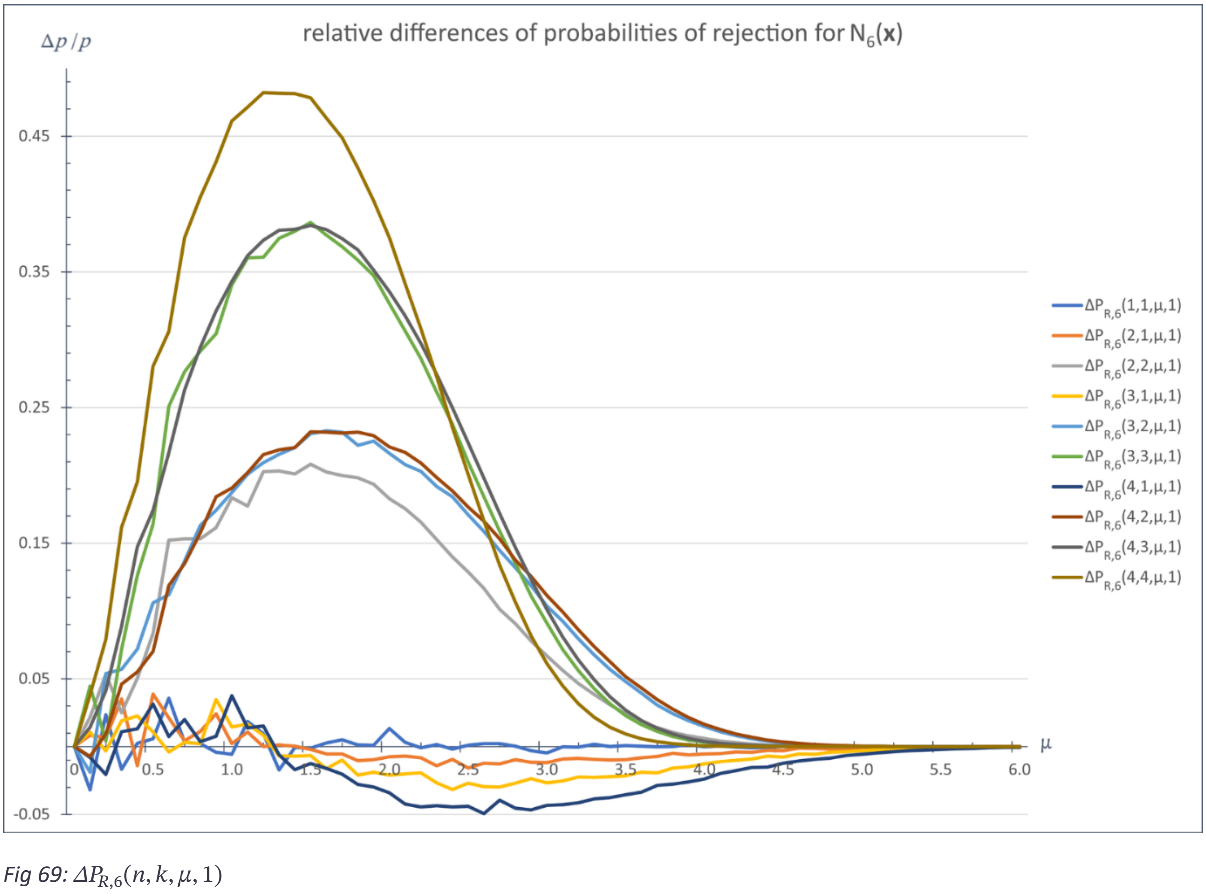

Fig 69: $\Delta P_{R,6}(n, k, \mu, 1)$

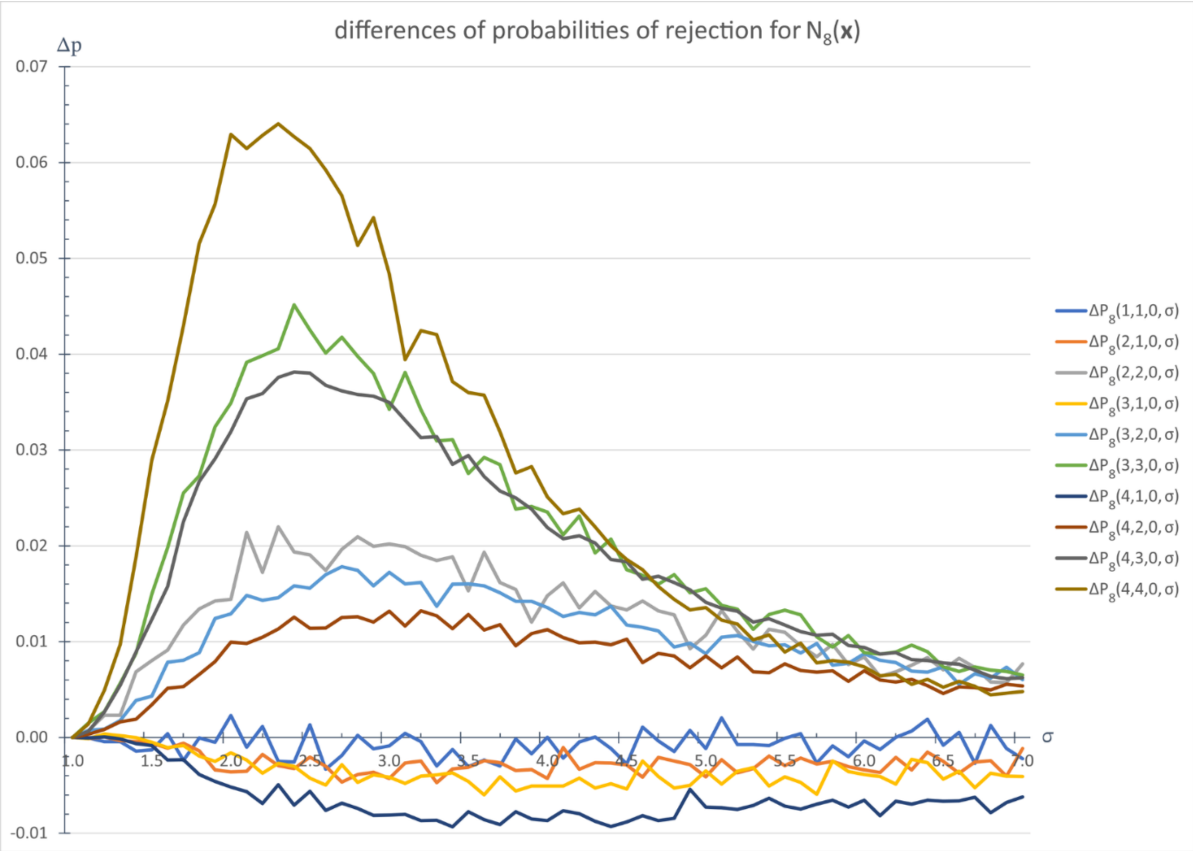

Fig 70: $\Delta P_8(n,k,0,\sigma)$

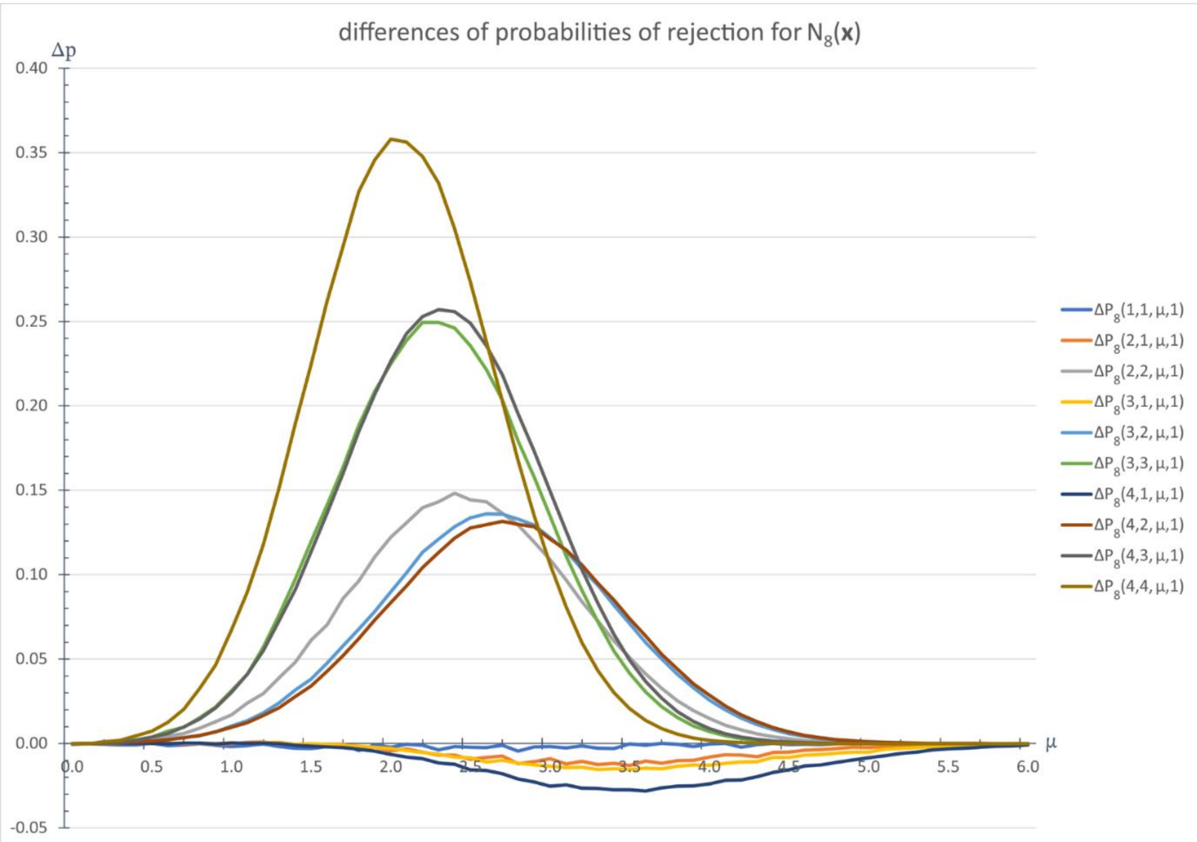

Fig 71: $\Delta P_8(n,k,\mu,1)$

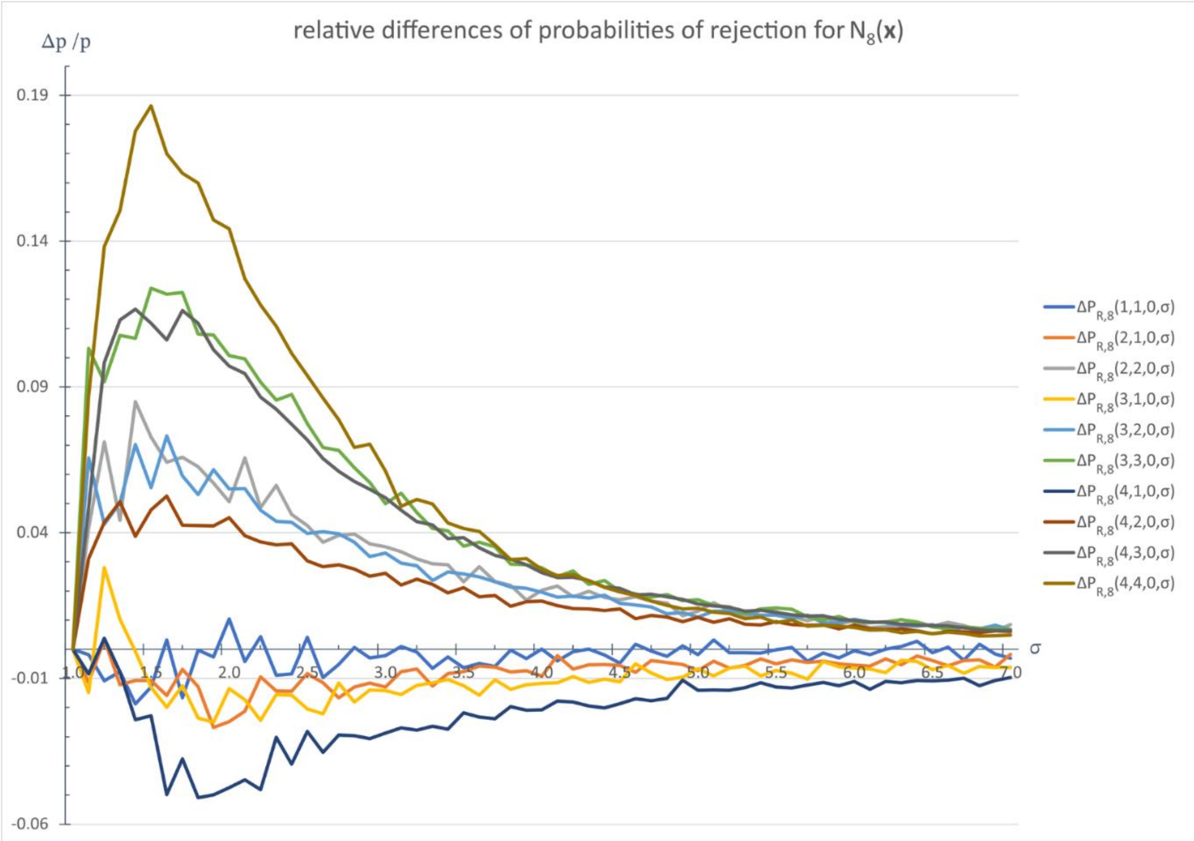

Fig 72: $\Delta P_{R,8}(n,k,0,\sigma)$

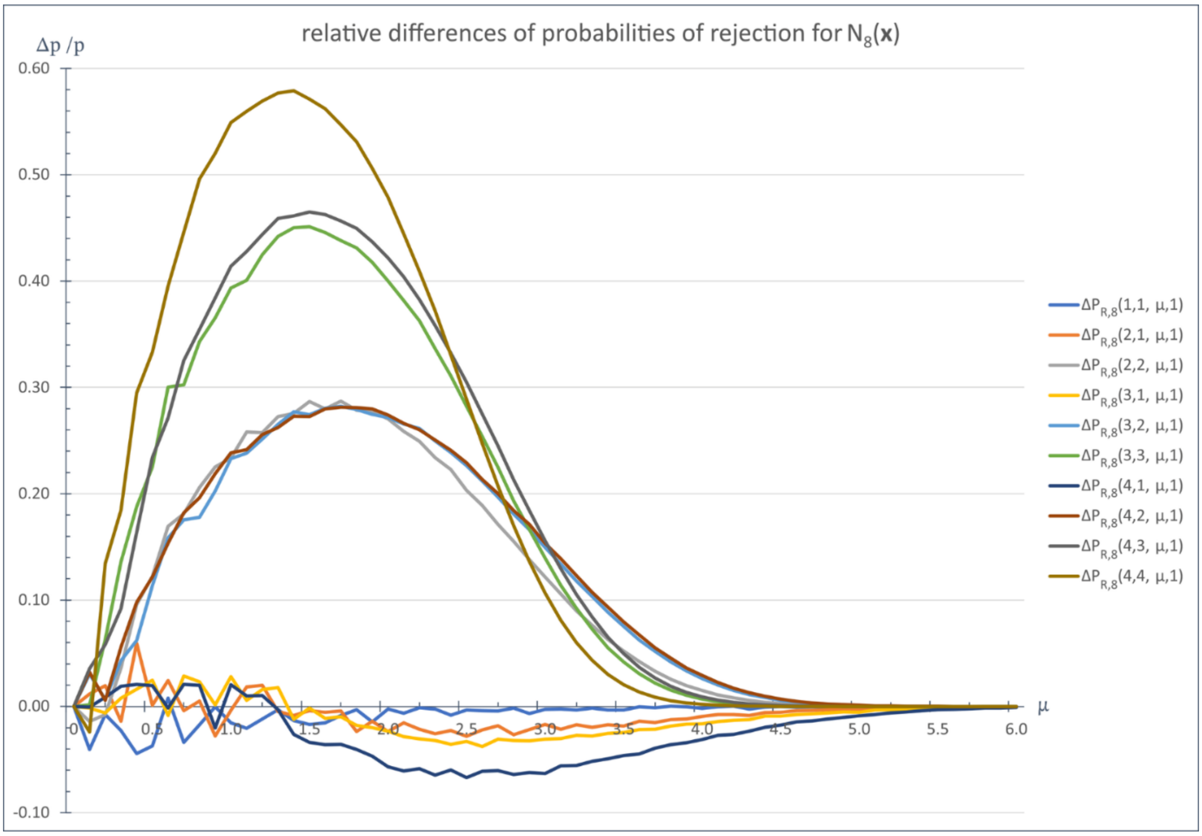

Fig 73: $\Delta P_{R,8}(n,k,\mu,1)$

Permanent Citation:

Chatzimichail RA, Hatjimihail AT. Quality Control Using Convolutional Neural Networks Applied to Samples of Very Small Size. Technical Report XXI. Drama: Hellenic Complex Systems Laboratory, 2022. Available at: https://www.hcsl.com/Documents/hcsltr21.pdf

License

First Published: June 18, 2022