Numerical Methods **Problem Set 1**
Professor: Cezar Santos **January 31, 2023**
TA: Rafael Vetromille **Student:** Luan Borelli

Preliminary comments. As requested, I have solved this problem set using two different programming languages: Julia and Matlab. Julia was the main language I used to work. After first solving the entire problem set in Julia, I translated the code into Matlab code with a few adaptations. The results presented in the next pages are the results obtained by the Julia code. I report the results obtained by the Matlab code in Appendix A. The results are roughly the same, the only difference is that the simulated AR processes are different. Both codes basically define five functions:

> `tauchen(`$\mu, \sigma^2, \rho, N, m$`)`: Given the parameters μ, σ^2 and ρ, the number of grid points N, and the maximum number of standard deviations from the mean, m, this function returns the states vector (grid points) and the transition matrix for the discrete Markov process approximation of an AR(1) process specified as $\theta_t = \mu(1-\rho) + \rho z_{t-1} + \epsilon_t$, $\epsilon_t \sim N(0, \sigma^2)$, by Tauchen's method.
>
> `rouwenhorst(`μ, σ, ρ, N`)`: Given the parameters μ, σ and ρ, and the number of grid points N, this function returns the states vector (grid points) and the transition matrix for the discrete Markov process approximation of an AR(1) process specified as $\theta_t = \mu(1-\rho) + \rho z_{t-1} + \epsilon_t$, $\epsilon_t \sim N(0, \sigma^2)$, by Rouwenhorst's method.
>
> `ar(`μ, σ, ρ, n`)`: Given the parameters μ, σ and ρ, and a sample size n, this function simulates an AR(1) process specified as $\theta_t = \mu(1-\rho) + \rho z_{t-1} + \epsilon_t$, $\epsilon_t \sim N(0, \sigma^2)$. It returns two outputs: the time series $\{\theta_t\}_{t=1}^n$ generated by the simulated autoregressive process and the series of errors $\{\epsilon_t\}_{t=1}^n$ used to generate the autoregressive process.
>
> `tauch_discretized_ar(`$\mu, \sigma^2, \rho, N, m, \{\epsilon_t\}_{t=1}^n$`)`: Given the parameters μ, σ^2, ρ, N, m, and a series of simulated errors (obtained from the `ar()` function output!), this function returns the Tauchen's discretized version of the continuous `ar()` time series — i.e., it returns a discretized time series of size n that approximates the continuous time series generated by the `ar()` function.
>
> `rouw_discretized_ar(`$\mu, \sigma^2, \rho, N, \{\epsilon_t\}_{t=1}^n$`)`: Given the parameters μ, σ^2, ρ, N, m, and a series of simulated errors (obtained from the `ar()` function output!), this function returns the Rouwenhorst's discretized version of the continuous `ar()` time series — i.e., it returns a discretized time series of size n that approximates the continuous time series generated by the `ar()` function.

Using these five functions I obtained the results presented in the next pages.[1]

---

[1] The results presented here are replicable. I've set a seed at the beginning of both Julia and Matlab codes that ensures the same results reported here will be obtained whenever the code is rerun.

1. *Discretize the process using the Tauchen (1986) method. Use 9 points.*

Solution. Defining the parameters $\mu = 0$, $\sigma = 0.007$, $\rho = 0.95$, $N = 9$ and $m = 3$ and running the function $\texttt{tauchen}(\mu, \sigma, \rho, N, m)$ I obtained:[2]

$$z_{\text{grid}}^{\text{Tauchen}} \approx$$

$$\begin{bmatrix} -0.06725382459813659 \\ -0.05044036844860244 \\ -0.03362691229906829 \\ -0.016813456149534146 \\ 0.0 \\ 0.016813456149534146 \\ 0.03362691229906829 \\ 0.05044036844860244 \\ 0.06725382459813659 \end{bmatrix} \tag{1}$$

$$\Pi^{\text{Tauchen}} \approx$$

$$\begin{bmatrix} 0.7644 & 0.2347 & 0.0009 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0592 & 0.7405 & 0.1997 & 0.0006 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0001 & 0.0747 & 0.7569 & 0.1679 & 0.0004 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0001 & 0.0931 & 0.7669 & 0.1396 & 0.0002 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0002 & 0.1147 & 0.7702 & 0.1147 & 0.0002 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0002 & 0.1396 & 0.7669 & 0.0931 & 0.0001 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0004 & 0.1679 & 0.7569 & 0.0747 & 0.0001 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0006 & 0.1997 & 0.7405 & 0.0592 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0009 & 0.2347 & 0.7644 \end{bmatrix} \tag{2}$$

□

[2] Transition matrix entries are rounded at the fourth decimal place. To be strict, the zeros in the matrix are actually numbers very close to zero.

2. *Discretize the process using the Rouwenhorst method. Use 9 points.*

Solution. Defining the parameters $\mu = 0$, $\sigma = 0.007$, $\rho = 0.95$ and $N = 9$ and running the function `rouwenhorst(`μ, σ, ρ, N`)` I obtained:[3]

$$z_{\text{grid}}^{\text{Rouwenhorst}} \approx \begin{bmatrix} -0.06340751391209735 \\ -0.04755563543407301 \\ -0.03170375695604868 \\ -0.01585187847802434 \\ 0.0 \\ 0.01585187847802434 \\ 0.03170375695604868 \\ 0.04755563543407301 \\ 0.06340751391209735 \end{bmatrix} \tag{3}$$

$$\Pi^{\text{Rouwenhorst}} \approx \begin{bmatrix} 0.8167 & 0.1675 & 0.015 & 0.0008 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0209 & 0.8204 & 0.1469 & 0.0113 & 0.0005 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0005 & 0.042 & 0.8231 & 0.1261 & 0.0081 & 0.0003 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0016 & 0.063 & 0.8247 & 0.1051 & 0.0054 & 0.0001 & 0.0 & 0.0 \\ 0.0 & 0.0001 & 0.0032 & 0.0841 & 0.8253 & 0.0841 & 0.0032 & 0.0001 & 0.0 \\ 0.0 & 0.0 & 0.0001 & 0.0054 & 0.1051 & 0.8247 & 0.063 & 0.0016 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0003 & 0.0081 & 0.1261 & 0.8231 & 0.042 & 0.0005 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0005 & 0.0113 & 0.1469 & 0.8204 & 0.0209 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0008 & 0.015 & 0.1675 & 0.8167 \end{bmatrix} \tag{4}$$

□

[3] Again, transition matrix entries are rounded at the fourth decimal place. To be strict, the zeros in the matrix are actually numbers very close to zero.

3. *Simulate the continuous process for 10000 periods. Do the same for the discretized processes. Compare the paths of each process.*

Solution. For this exercise, I first simulated an AR(1) process using the function `ar`(μ, σ, ρ, n) with $n = 10000$. Then, I used the errors $\{\epsilon_t\}_{t=1}^{10000}$ returned by this function to approximate the simulated AR(1) by both Tauchen's and Rouwenhorst's methods, using the functions `tauch_discretized_ar`$(\mu, \sigma^2, \rho, N, m, \{\epsilon_t\}_{t=1}^{n})$ and `rouw_discretized_ar`$(\mu, \sigma^2, \rho, N, \{\epsilon_t\}_{t=1}^{n})$. Finally, I plotted both the simulated continuous AR(1) and the approximated discretized processes together. The plots are reported in Figures 1, 2, and 3.[4]

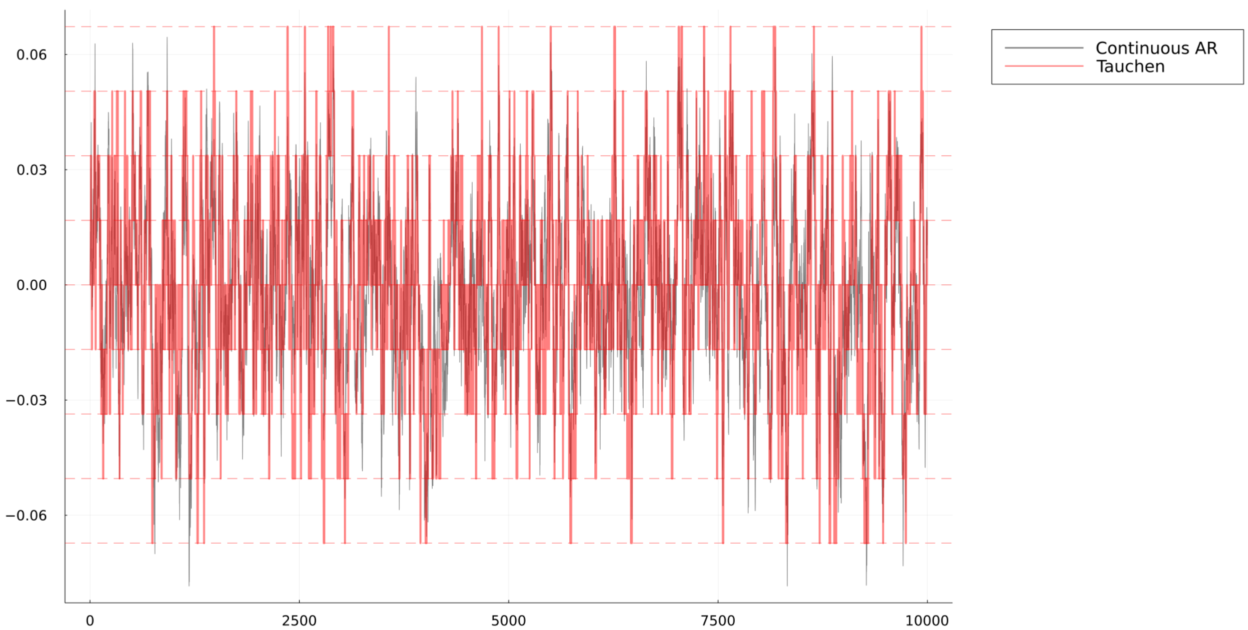

Figure 1: Continuous AR(1) and its approximation by Tauchen's discretization.

[4] As can be noticed, it is somewhat difficult to compare the continuous and approximate process in the graphs presented here. This is due to the sample size ($n = 10000$) which is too large to allow a clear visualization of the overlapping time series. I tried resizing the scale of the plots to make viewing easier, but even with a few tweaks I couldn't improve the situation much. Out of curiosity, to facilitate the visualization of how the approximations are being made by both methods, I report in Appendix B some graphs with a reduced sample size ($n = 1000$).

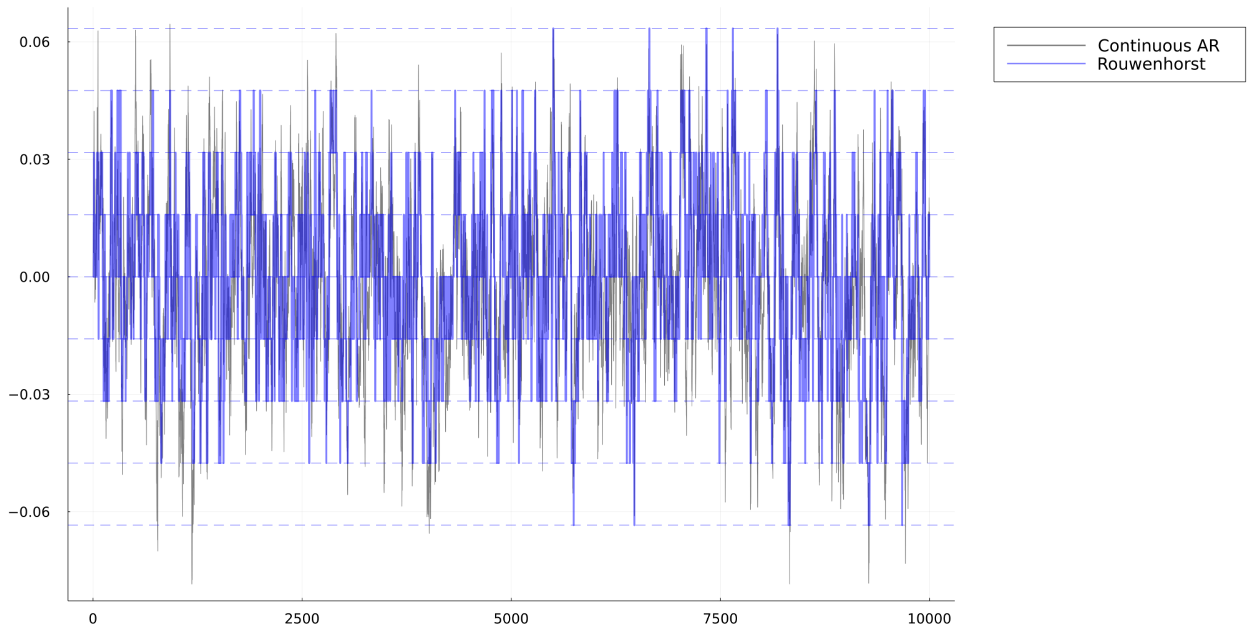

Figure 2: Continuous AR(1) and its approximation by Rouwenhorst's discretization.

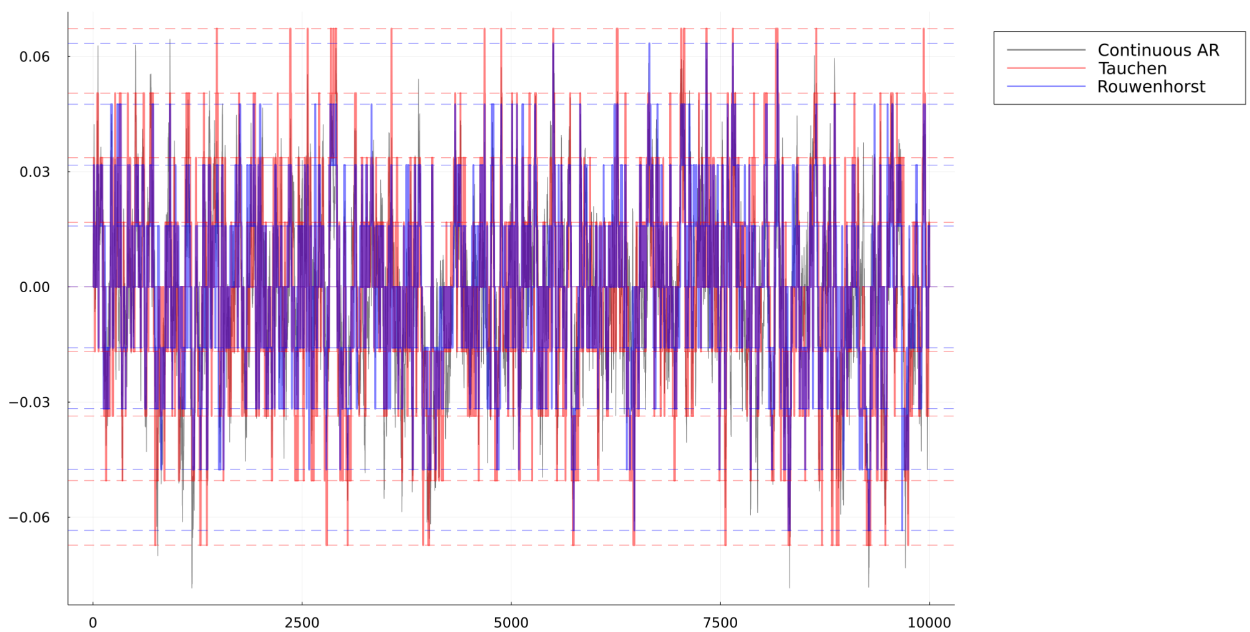

Figure 3: Continuous AR(1) and its approximation by both methods.

As a bonus, to assess how good each method's approximations were, I calculated the Mean Squared Errors (MSQE) for each case and also analyzed the relative MSQE. Note that a relative MSQE greater than unity would indicate that Rouwenhorst's method performed better than Tauchen's method, while less than unity would indicate that Tauchen's method performed better than Rouwenhorst's method. The results are presented in Table 3.

Table 1: Mean Squared Error (MSQE) of each approximation method.

MSQE Tauchen	**MSQE Rouwenhorst**	**Relative MSQE**
0.02233787652155521	0.02329290966441272	0.9589989762285205

Observe that both methods seem to approximate the continuous AR(1) process reasonably well (both have quite low absolute MSQEs). Furthermore, the Relative MSQE is quite close to unity, but still below, indicating that the Tauchen's method is performing slightly better than the Rouwenhorst's method for this particular case.

□

---

4. *Estimate the AR(1) processes based on simulated data, both for Tauchen and Rouwenhorst. How close are they form the real data generating process?*

Solution. Using the Julia package `ARCHModels`, I estimated an AR(1) model using as inputs the approximate processes obtained in the previous question for each discretization method. The results are presented in Table 2.

Table 2: AR(1) estimates for the discretized approximations.

	$\hat{\rho}$	**Error** $(\lvert\rho - \hat{\rho}\rvert)$
Tauchen	0.9493020452072056	0.0006979547927943308
Rouwenhorst	0.9474712354662483	0.002528764533751615

As can be seen, the errors for both coefficient estimates with respect to the actual value of the coefficient ($\rho = 0.95$) are very small, which indicates that the approximations obtained by both methods are quite good. It is interesting to note that Tauchen's method has a smaller error than Rouwenhorst's method. Also note that the results obtained here are in line with the previous results we obtained when calculating the MSQEs in item 3, where we concluded that Tauchen's method performed slightly better than Rouwenhorst's method.

---

5. *Redo the questions above with $\rho = 0.99$.*

Solution.

(1)

$$z_{\text{grid}}^{\text{Tauchen}} \approx$$

$$\begin{bmatrix} -0.14886505305175046 \\ -0.11164878978881285 \\ -0.07443252652587523 \\ -0.037216263262937616 \\ 0.0 \\ 0.037216263262937616 \\ 0.07443252652587523 \\ 0.11164878978881285 \\ 0.14886505305175046 \end{bmatrix} \tag{5}$$

$$\Pi^{\text{Tauchen}} \approx$$

$$\begin{bmatrix} 0.9928 & 0.0072 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0024 & 0.9914 & 0.0062 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0028 & 0.9918 & 0.0054 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0033 & 0.9921 & 0.0046 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0039 & 0.9921 & 0.0039 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0046 & 0.9921 & 0.0033 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0054 & 0.9918 & 0.0028 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0062 & 0.9914 & 0.0024 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0072 & 0.9928 \end{bmatrix} \tag{6}$$

(2)

$$z_{\text{grid}}^{\text{Rouwenhorst}} \approx$$

$$\begin{bmatrix} -0.14035131799278386 \\ -0.10526348849458789 \\ -0.07017565899639193 \\ -0.035087829498195965 \\ 0.0 \\ 0.035087829498195965 \\ 0.07017565899639193 \\ 0.10526348849458789 \\ 0.14035131799278386 \end{bmatrix} \tag{7}$$

$$\Pi^{\text{Rouwenhorst}} \approx$$

$$\begin{bmatrix} 0.9607 & 0.0386 & 0.0007 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0048 & 0.9609 & 0.0338 & 0.0005 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0097 & 0.961 & 0.029 & 0.0004 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0001 & 0.0145 & 0.9611 & 0.0241 & 0.0002 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0001 & 0.0193 & 0.9611 & 0.0193 & 0.0001 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0002 & 0.0241 & 0.9611 & 0.0145 & 0.0001 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0004 & 0.029 & 0.961 & 0.0097 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0005 & 0.0338 & 0.9609 & 0.0048 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0007 & 0.0386 & 0.9607 \end{bmatrix} \tag{8}$$

(3)

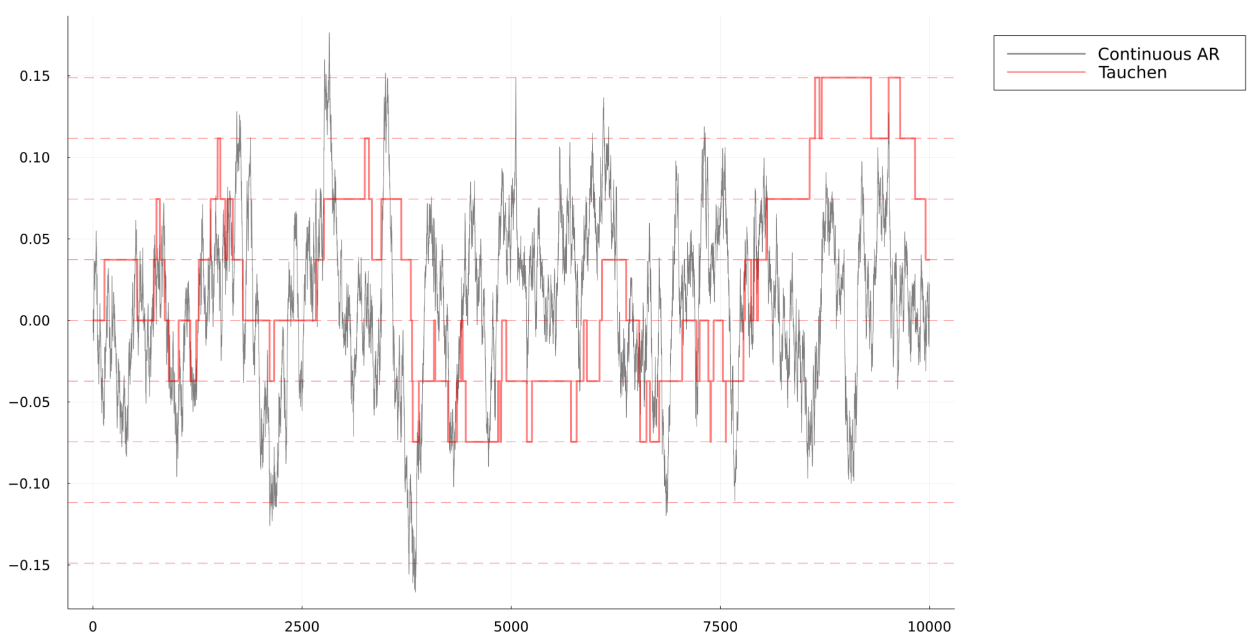

Figure 4: Continuous AR(1) and its approximation by Tauchen's discretization, $\rho = 0.99$.

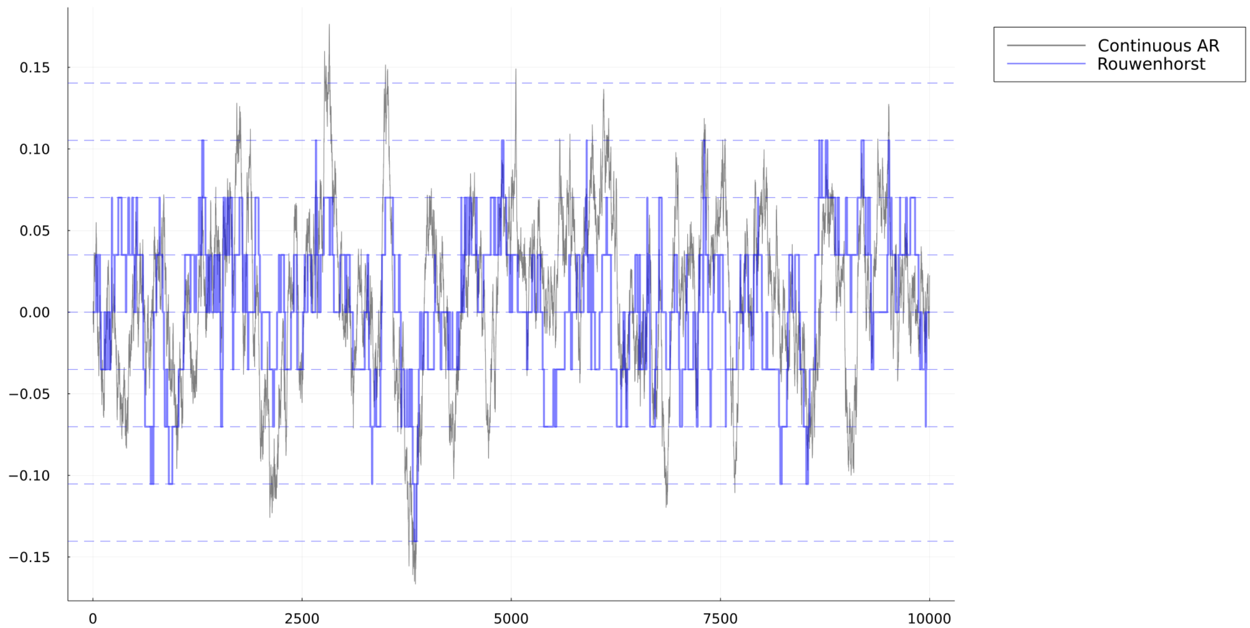

Figure 5: Continuous AR(1) and its approximation by Rouwenhorst's discretization, $\rho = 0.99$.

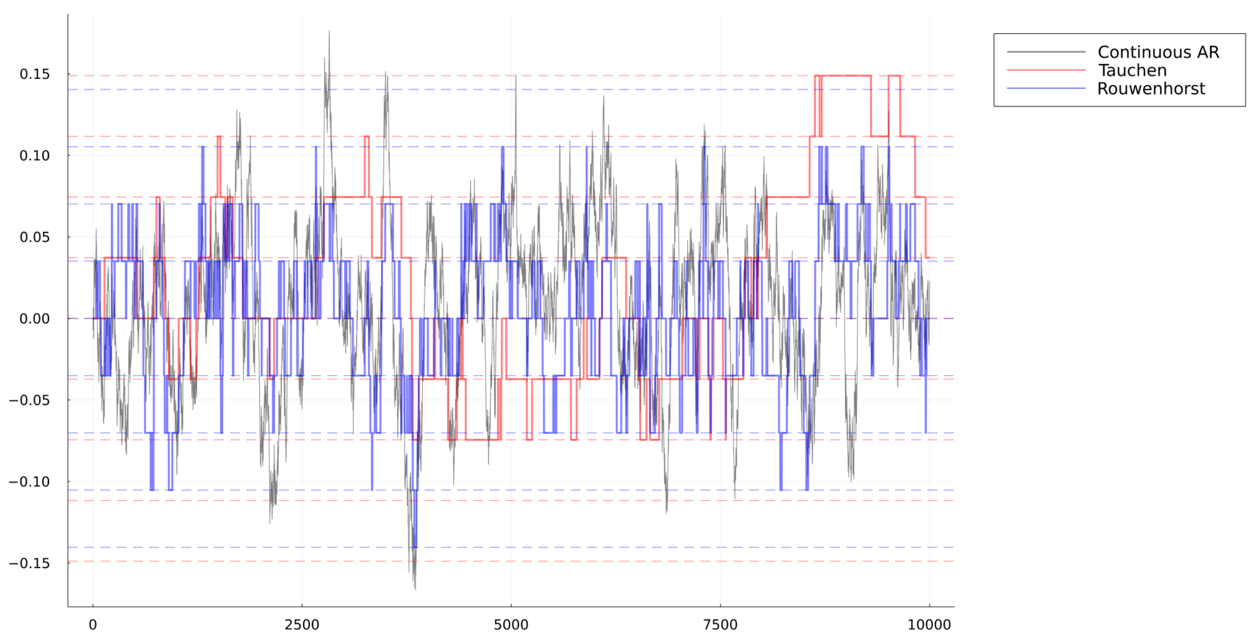

Figure 6: Continuous AR(1) and its approximation by both methods, $\rho = 0.99$.

Table 3: Mean Squared Error (MSQE) of each approximation method, $\rho = 0.99$.

MSQE Tauchen	**MSQE Rouwenhorst**	**Relative MSQE**
0.5930064309988402	0.25583018573928107	2.3179689655667866

As can be seen, with $\rho = 0.99$ the approximations become much worse for both methods when compared to the approximations we obtained earlier for the case of $\rho = 0.95$ (to see this, compare the absolute MSQEs that we got now with the absolute MSQEs obtained earlier, in Table 2). This indicates that for a higher degree of persistence, both methods have more difficulties to approximate the real process. When comparing between methods, it can be noticed that Tauchen's method now performed remarkably poorly, while Rouwenhorst's method performed (relatively) much better when compared to Tauchen's. Indeed, the Relative MSQE tells us that Tauchen's method had an MSQE more than twice (!) higher than Rouwenhorst's method. This is in line with what the theory would predict: Rouwenhorst's method is expected to be better than Tauchen's method when the persistence of the autoregressive process is very high. As $\rho \to 1$, the performance of Rouwenhorst's method dominates the performance of Tauchen's method.

(4)

Table 4: AR(1) estimates for the discretized approximation, $\rho = 0.99$.

	$\hat{\rho}$	**Error** $(\lvert\rho - \hat{\rho}\rvert)$
Tauchen	0.9985436627125617	0.008543662712561684
Rouwenhorst	0.9867180393094456	0.0032819606905544196

As can be seen, when compared with the estimates obtained in Table 2, the estimation errors became larger for both methods. This again indicates that for a higher degree of persistence, both methods have more difficulties to approximate the real process. Furthermore, it is possible to notice that now Rouwenhorst's estimate is closer to the real value of ρ than Tauchen's estimates. As already discussed, this is in line with the theory and corroborates the evidence previously pointed out by the MSQEs (that Rouwenhorst's method is expected to be better than Tauchen's method when the persistence of the autoregressive process is very high).

One way of trying to improve the approximations for this case of very high persistence is to increase the number of grid points considered in the discretization processes. In Appendix C I report results considering $N = 20$ grid points for the case $\rho = 0.99$. Approximations become much better.

Appendix A Matlab Results

The grid points and transition probabilities matrices obtained by the Matlab code were exactly the same as that obtained by the Julia code and reported in equations (1), (2), (3) and (4), so I will omit them here. The simulated process together with the discretized approximations obtained by the Matlab code is reported in the figure below, for both cases $\rho = 0.95$ and $\rho = 0.99$.

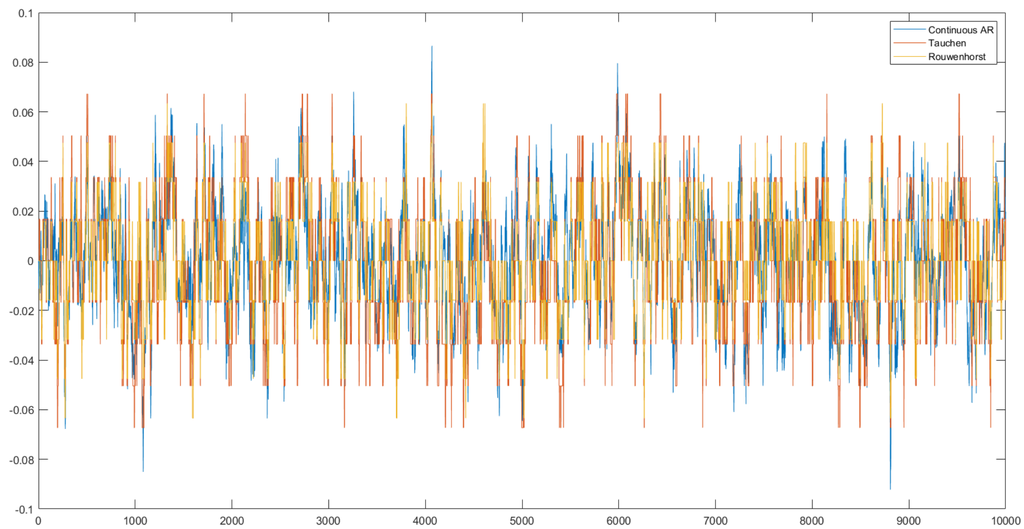

Figure 7: Continuous AR(1) and its approximation by both methods, $\rho = 0.95$.

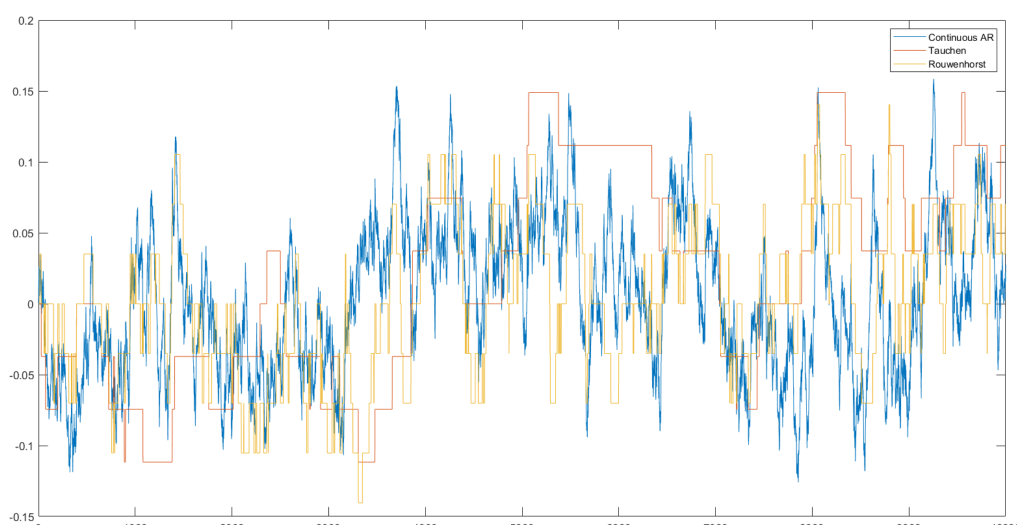

Figure 8: Continuous AR(1) and its approximation by both methods, $\rho = 0.99$.

The AR(1) estimates for each discretized process in each case are reported in the tables below.

Table 5: Matlab results: AR(1) estimates for the discretized approximation, $\rho = 0.95$.

	$\hat{\rho}$	**Error** $(\lvert\rho - \hat{\rho}\rvert)$
Tauchen	0.9473	0.0027
Rouwenhorst	0.9456	0.0044

Table 6: Matlab results: AR(1) estimates for the discretized approximation, $\rho = 0.99$.

	$\hat{\rho}$	**Error** $(\lvert\rho - \hat{\rho}\rvert)$
Tauchen	0.9992	0.0092
Rouwenhorst	0.9921	0.0021

As can be seen, similarly to Julia's results, Tauchen's method performed better than Rouwenhorst's in the case $\rho = 0.95$, while Rouwenhorst's method performed much better than Tauchen's method in the case $\rho = 0.99$.

Appendix B $n = 1000$ for a better visualization

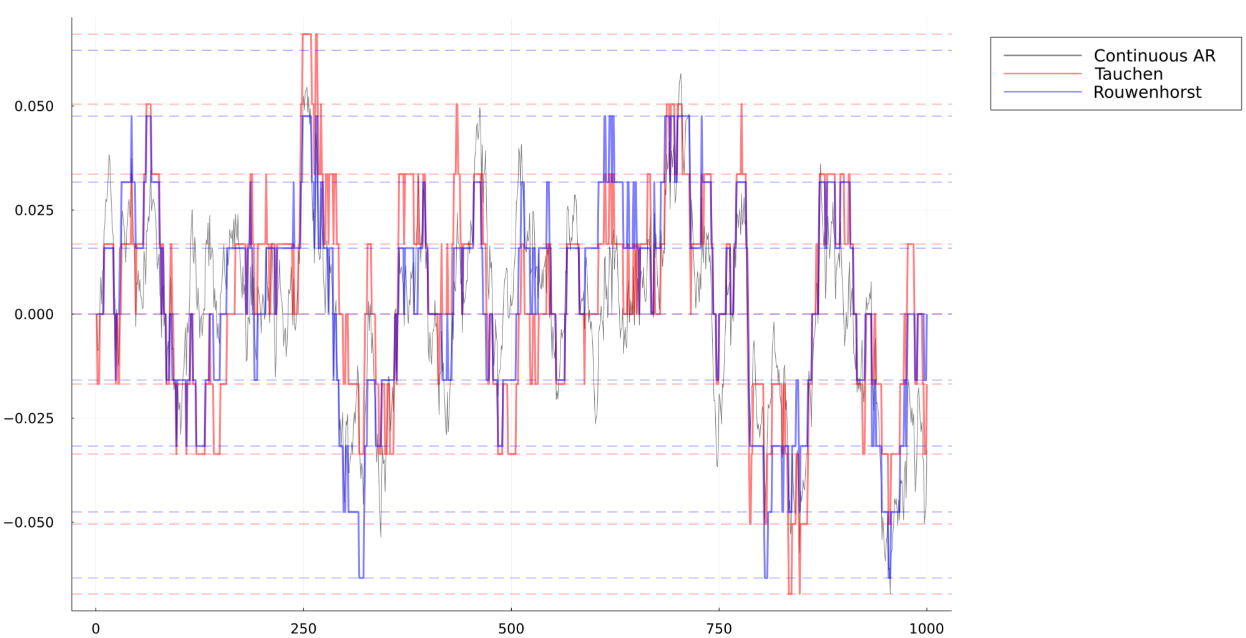

Figure 9: Continuous AR(1) and its approximation by both methods, $\rho = 0.95$.

Appendix C More grid points for the $\rho = 0.99$ case

Setting the number of grid points for discretization to $N = 20$ we obtain the following results:

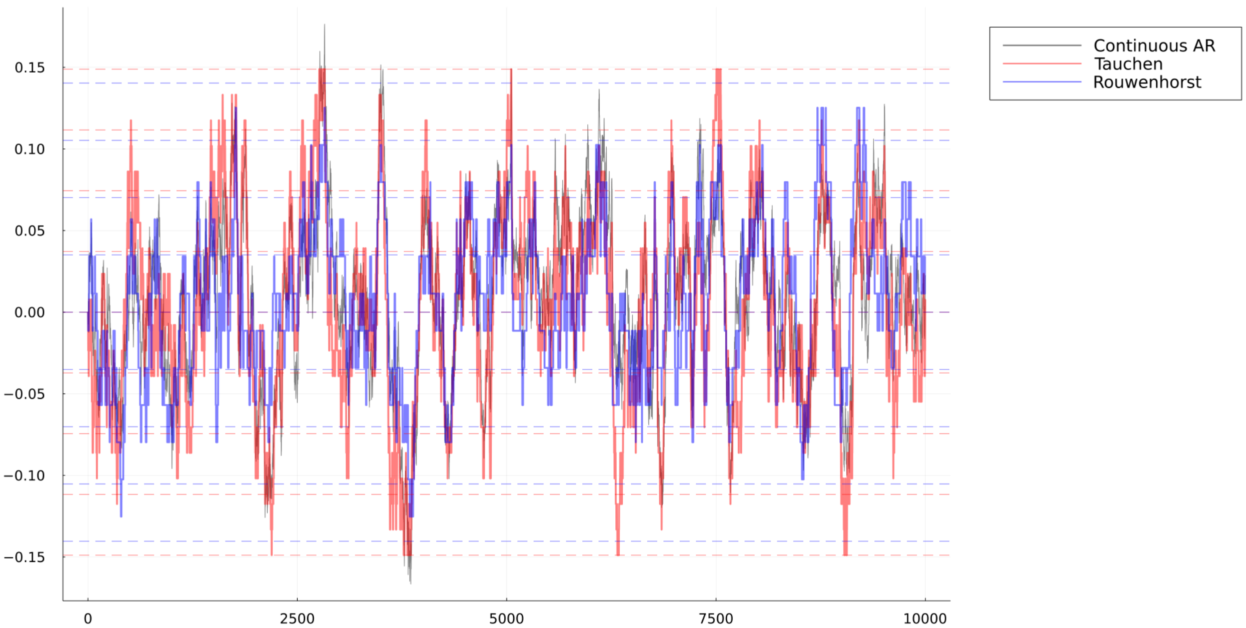

Figure 10: Continuous AR(1) and its approximation by both methods, $\rho = 0.95$ with $N = 20$.

As can be seen, the approximations became much better when compared to the approximations obtained with $N = 9$ grid points. This indicates that, in general, for highly persistent processes it is desirable to use a higher number of grid points for discretization.