Modalitati de estimare a parametrilor modelelor ARMA

Ec. Adrian HOSPODAR, Romtelecom
e-mail: ahospoda@fx.ro

Articolul prezinta doi algoritmi utilizati pentru estimarea parametrilor modelelor ARMA. Prezentarea teoretica este completata de prezentarea modului în care s-a facut implementarea celor doi algoritmi ca metoda în biblioteca ECONO. Performantele numerice sunt analizate atât din punct de vedere teoretic cât si comparativ cu alte programe.
***Cuvinte cheie**: estimare, parametri, model ARMA, analiza econometrica.*

1 Introducere

Estimarea parametrilor modelelor Auto Regressive Moving Average (ARMA pe scurt) este astazi o problema rezolvata. Exista mai multi algoritmi care pot fi folositi în acest scop. Implementari ale acestor algoritmi sunt oferite în orice pachet de programe destinat analizei seriilor de timp.

Diferentierile dintre acesti algoritmi sunt numeroase. O parte dintre ei folosesc ca metoda de estimare metoda celor mai mici patrate, în diferite variante de implementare. Alti algoritmi folosesc metoda verosimilitatii maxime ca metoda de estimare. Un prim punct de interes al acestui articol este de a prezenta doi algoritmi din ultima categorie.

Un al doilea punct de interes este de a prezenta metodele numerice folosite de cei doi algoritmi. Metodele numerice sunt folosite deoarece forma functiei de verosimilitate nu permite în general obtinerea de solutii analitice. În prezentarea acestor metode numerice accentul va fi pus pe calitatile estimatorilor obtinuti.

Cel de-al treilea punct de interes al articolului este de a prezenta o modalitate concreta de implementare a algoritmilor prezentati. Accesul la implementari se face prin apelarea unei functii din biblioteca ECONO. În prezentarea modalitatii de implementare accentul se va pune pe modul în care estimatorii obtinuti sunt folositi ca date de intrare în alte functii ale bibliotecii.

Organizarea articolului este urmatoarea: prima sa sectiune prezinta din punct de vedere teoretic cei doi algoritmi folositi în mod curent pentru estimarea parametrilor modelelor ARMA. Cea de a doua sectiune prezinta pe scurt functiile bibliotecii ECONO care folosesc drept date de intrare parametrii modelelor ARMA. Cea de a treia sectiune prezinta modalitatea concreta de implementare în biblioteca ECONO a acestor algoritmi. Ultima sectiune prezinta câteva din performantele numerice ale implementarii existente în biblioteca ECONO, comparativ cu alte pachete de programe.

2. Metode de estimare a parametrilor modelelor ARMA

Modelul ARMA se defineste astfel:

$$\Phi(B)\nabla^d x_t = q_0 + q(B)a_t \quad (1)$$

unde:

x_t - variabila observata la momentul t (posibil o transformare a acesteia ce este cunoscuta)

d - un numar întreg, pozitiv, finit;

∇ -operatorul de diferentiere, ales astfel încât $\nabla^d x_t$ este stationara;

Φ -polinom de grad p ai carui coeficienti reprezinta partea autoregresiva a modelului ARMA;

θ - un polinom de grad q ai carui coeficienti reprezinta partea de medie mobila a modelului ARMA;

a_t, t=1..T - secventa de numere aleatore independente si distribuite normal cu media 0 si dispersia finita σ^2 (~ N(0, σ) pe scurt);

T – numarul de date observate.

Se presupun cunoscuti parametrii d, p si q. Problema estimarii parametrilor modelelor ARMA consta în determinarea coeficientilor polinoamelor Φ si θ, notati cu Φ=(Φ_1,

$\Phi_2,\ldots,\Phi_p)'$ si $\theta=(\theta_1,\theta_2,\ldots,\theta_q)'$, si la determinarea dispersiei σ^2 a variabilei a_t. Se vor utiliza urmatoarele notatii: $w=\nabla^d x$, $\omega=(w,\Phi,\theta,\sigma^2)$.

Modelul ARMA din ecuatia (1) se rescrie astfel:

$$a_t=\theta_0+\theta_1*a_{t-1}+\ldots+\theta_q*a_{t-q}+w_t-\phi_1*w_{t-1}-\ldots-\phi_p*w_{t-p} \quad (2)$$

Estimarea vectorului necunoscutelor ω se poate face prin utilizarea de algoritmi ce se bazeaza fie pe metoda celor mai mici patrate, fie pe cea a verosimilitatii maxime. Algoritmii prezentati în continuare folosesc cea de a doua metoda. Pentru un algoritm bazat pe metoda celor mai mici patrate, vezi [2], pag. 200-205.

Primul pas în construirea unui algoritm ce utilizeaza metoda verosimilitatii maxime este scrierea functiei de verosimilitate a parametrilor modelului.

Deorece se presupune ca $a_t \sim N(0,\sigma)$ functia de verosimilitate a parametrilor ce trebuie aflati (Φ,θ,σ^2) este aceeasi cu functia de probabilitate compusa a w data de ecuatia (3).

$$L(\phi,\theta,\sigma^2/w)=\prod_{t=1-p-q}^{T}(\phi_1*w_{t-1}+\ldots+\phi_p*w_{t-p}-\theta_0-\theta_1*a_{t-1}-\ldots\theta_q*a_{t-q}+a_t)$$

$$L(\phi,\theta,\sigma^2/w)=(2*\pi*d_w^2)^{\frac{p+q-1}{2}}*e^{-\frac{1}{2*d_w^2}*\sum_{t=1-p-q}^{T}(w_i-\overline{w})^2} \quad (3)$$

unde w este media $\{w_t\}_{t=1..T}$, iar d_w^2 este dispersia $\{w_t\}_{t=1..T}$

Functia de verosimilitate (3) depinde de w si d_w^2. Forma ei depinde de ordinul modelului ARMA si poate fi destul de complicata. De exemplu, pentru ARMA (1,1), media este data de (3a) iar dispersia are o forma complicata.

$$\overline{w}=\frac{\theta_0}{\phi_1-1} \quad (3a)$$

Pornind de la (3) functia de verosimilitate maxima se rescrie astfel încât sa fie valabila pentru orice model ARMA. Pentru aceasta vom folosi doua functii g_1 si g_2 care exprima dependenta între vectorul necunoscutelor si media, respectiv dispersia variabilei w_t. Functia de verosimilitate scrisa în acest fel este data de (4).

$$L(\phi,\theta,\sigma^2/w)=g_1(\phi,\theta,\sigma^2)*e^{-\frac{1}{2\sigma^2}*S(\phi,\theta)} \quad (4)$$

unde

$$S(\phi,\theta)=\sum_{t=1-p-q}^{T}E(u_t/w)^2 \quad (5)$$

g_1 este o functie de parametrii (Φ,θ,σ^2)

În ecuatia (5) simbolurile nou introduse au urmatoarea semnificatie:

$$u_t=\begin{cases} a_t, & t=1,2,\ldots T \\ g_2(a^*,w^*), & t\le 0 \end{cases} \quad (5)$$

$E(u_t/\omega)$ este speranta matematica a variabilei u_t conditionata de vectorul parametrilor ω.

g_2 este o functie liniara de valorile initiale neobservate a^*, ω^*. Aceste valori sunt necesare pentru a calcula pe a_t în (2). Ele vor fi folosite doar la initializarea algoritmului. O implementare flexibila a algoritmului trebuie sa ofere atât valori prestabilite, cât si posibilitatea de a modificare a lor.

Forma functiilor g_1 si g_2 depind de ordinul modelului. Interesul trebuie sa se concentreze asupra formei functiei g_2, deoarece estimarea parametrilor se face, de regula maximizând forma redusa a functiei de verosimilitate (4), în care se neglijeaza influenta data de g_1. Motivul acestui mod de lucru este dat de forma complicata pe care o are functia de verosimilitate (4) si de dificultatea de gasire a maximului global. Forma functiei ce trebuie maximizata este data de (6).

$$L(\phi,\theta,\sigma^2/w)=e^{-\frac{1}{2\sigma^2}*S(\phi,\theta)} \quad (6)$$

Maximizarea lui (6) este echivalenta cu minimizarea lui $S(\phi,\theta)$. Estimatorii sunt obtinuti prin metoda celor mai mici patrate si au proprietatile obisnuite.

Pentru ca aceasta estimare sa fie posibila, trebuie însa calculate $E(u_t/\omega)$ (t=1-p-q, …, 0). Aceasta presupune calculul lui $E(a_t/\omega)$ (t=1-q, …, 0) si $E(w_t/\omega)$ (t=1-p,…,0). Modul în care acest calcul este facut conduce la doi algoritmi diferiti.

Prima modalitate de estimare a lui $E(u_t/\omega)$ este de a adapta metoda folosita în mod curent pentru efectuarea de prognoze cu un model ARMA. Metoda de generare de prognoze foloseste faptul ca cea mai buna prognoza ce se poate efectua la momentul n pentru un orizont de prognoza de m unitati de timp este data de speranta matematica a lui w_{n+m} conditionata de valorile trecute (si cunoscute) w_n, w_{n-1},…. Notatiile folosite sunt cele din ecuatia (2). Formalizarea este data de (7).

$$p_{n,m} = E(w_{n+m} / w_n, w_{n-1}, \ldots) \quad (7)$$

Pentru a calcula $E(a_t/\omega)$ si $E(w_t/\omega)$ se construieste un nou model ARMA (p,q) dat de ecuatia (8).

$$e_t = q_0 + q_1 * e_{t+1} + \ldots + q_q * e_{t+q} + w_t - f_1 * w_{t+1} - \ldots - f_p * w_{t+p} \quad t = 1 - p - q, \ldots, 0 \quad (8)$$

Asemanarea cu modelul ARMA, ai carui parametri trebuie estimati, este evidenta. Modelul dat de ecuatia (8) exprima însa dependenta între valorile observate la momentul prezent si cele posterioare lui. Din aceasta cauza, modelul ARMA (8) este denumit si model ARMA *înapoi*.

Valorile autocorelatiei lui w_t exprimat ca în (8) sunt egale cu valorile autocorelatiei lui w_t exprimat ca în (2). În plus, dispersia lui e_t din (8) este aceeasi cu dispersia termenului eroare a_t din (2). De aceea, efectuarea de "previziuni" cu modelul (8) permite obtinerea lui $E(a_t/\omega)$ (t=1-q, …, 0) si $E(w_t/\omega)$ (t=1-p,…,0).

Odata ce aceste estimari sunt obtinute, vectorul parametrilor ω ce minimizeaza $S(\phi,\theta)$ este determinat prin algoritmul celor mai mici patrate neliniar.

Metoda este denumita *"backforecasting method"* datorita semnificatiei modelului (8). Alta denumire curent utilizata a metodei este *"metoda McLeod"* , dupa numele celui care a propus-o.

Cea de a doua modalitate de estimare a $E(a_t/\omega)$ (t=1-q, …, 0) si $E(w_t/\omega)$ (t=1-p,…,0) se bazeaza pe faptul ca speranta matematica a termenului eroare a_t din modelul (2) este egala cu 0. Astfel, se initializeaza valorile initiale ale termenului eroare a_t, t=p..p+1-q cu zero. Parametrii modelului ARMA sunt estimati apoi minimizând suma $S_c(\phi,\theta)$. Minimizarea se face cu algoritmul celor mai mici patrate neliniar.

Metoda este denumita *"metoda celor mai mici patrate conditionate"*.

Aplicarea oricarei metode din cele doua duce la obtinerea de estimatori consistenti, asimptotic eficienti si normal repartizati. Matricea covariantelor estimatorilor este asimptotic echivalenta cu matricea covariantelor estimatorilor calculata de algoritmul celor mai mici patrate neliniar.

Sunt necesare câteva observatii referitoare la deosebirile între calitatile estimatorilor obtinuti prin cele doua metode:

- Îndepartarea termenului g_1 din (5) nu modifica semnificativ functia de verosimilitate în cazul în care valoarea parametrilor modelului în modul este apropiata de unu. În acest caz functia de verosimilitate este aproximativ egala cu functia de verosimilitate redusa data de (6). În acest caz, metoda backforecasting asigura o convergenta mai rapida.
- În cazul în care dimensiunea seriei de timp analizata este mica si valoarea absoluta a parametrilor modelului nu se apropie de unu, metoda celor mai mici patrate conditionate este recomandata.
- În ceea ce priveste dispersia termenului eroare experienta numerica arata ca metoda *backforecasting* îl subestimeaza, iar metoda celor mai mici patrate conditionate îl supraestimeaza.

Pe baza acestor observatii, metoda celor mai mici patrate conditionate este preferabila metodei backforecasting. Dezavantajul acesteia, care apare atunci când valoarea parametrilor modelului în modul este apropiata de unu, poate fi depasit prin utilizarea unui algoritm performant cu o implementare flexibila, care sa furnizeze valori de start si sa permita modificarea lor de catre utilizator.

3. Prezentarea bibliotecii de functii ECONO.

Biblioteca ECONO ([5] pentru detalii) ofera accesul la o colectie de proceduri si algoritmi utilizati în analiza econometrica a seriilor de timp. Deoarece este proiectata si programata prin utilizarea paradigmei obiectelor, ea este organizata în clase. În versiunea 2.04, biblioteca contine urmatoarele clase:

CTmeS - este clasa de baza a bibliotecii. Ea ofera functii pentru calculul unor proprietati statistice ale seriei de timp si pentru transformarea initiala a datelor. Transformarea initiala a datelor se poate face asistat sau neasistat.

CArma - contine functii de transformare preliminara a datelor pentru modelarea prin modele ARMA, functii de specificare si estimare a modelelor si functii de estimare a calitatii procesului de modelare.

CRezid - contine functii ce calculeaza proprietati statistice specifice valorilor reziduale ce rezulta din estimarea unui model teoretic.

CTar - contine functii de specificare si estimare a modelelor din clasa TAR si functii de estimare a calitatii procesului de modelare.

CBayes - contine functii de calcul al verosimilitatilor marginale si de construire a portofoliilor de modele.

Reprezentarea grafica a ierarhiei claselor este prezentata în figura 1.

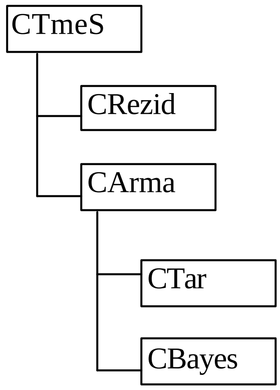

Fig.1. Ierarhia claselor bibliotecii ECONO

Estimarea modelelor ARMA este implementata ca o metoda a clasei ARMA. Rezultatele obtinute în urma aplicarii metodei sunt folosite de alte metode ale clasei. Totodata, datorita structurii ierarhice a bibliotecii ea este utilizata si în clasa CTar. Aceasta face ca de calitatea rezultatelor obtinute în urma aplicarii metodei ce implementeaza estimarea modelelor ARMA sa depinda calitatea rezultatelor obtinute prin aplicarea altor metode, din clasa CArma sau CTar.

Pentru a evidentia importanta acestei metode, vor fi prezentate în tabelul 1 metodele din clasele CArma si CTar ce folosesc drept date de intrare estimatiile parametrilor modelelor ARMA.

Tabel 1. Metodele ce folosesc ca date de intrare estimatiile parametrilor modelelor ARMA

Clasa	**Denumire metoda**	**Explicatii**
CArma	ArmaInAr	Metoda ofera o reprezentare pur autoregresiva a unei serii de timp pornind de la reprezentarea sa printr-un model ARMA. Determinarea valorilor coeficientilor modelului autoregresiv se face pe cale analitica pornind de la estimarile parametrilor modelului ARMA. De aceea, calitatea rezultatelor metodei depinde în întregime de calitatea estimarilor parametrilor modelului ARMA.
CArma	ArmaInMa	Metoda ofera o reprezentare printr-un model MA al unei serii de timp pornind de la reprezentarea sa printr-un model ARMA. Observatiile de la metoda ArmaInAr ramân valabile.

CArma	AlegeModel	Metoda selecteaza automat ordinul unui model ARMA folosind unul din criteriile informationale AIC, BIC sau Schwarz. Selectarea se face utilizând dispersia termenului eroare.
CTar	Constructor	Clasa CTar este derivata direct din CArma. Pe scurt, un model TAR este format din mai multe modele ARMA si un mecanism de tranzitie de la un model ARMA la altul. Fiecare din aceste modele ARMA trebuie estimat.

4. Implementarea algoritmilor în biblioteca ECONO

Trei observatii sunt necesare înainte de a prezenta modalitatea de implementare a metodelor descrise teoretic în prima sectiune a articolului.

Prima observatie este ca termenul de *algoritm c.m.m.p. neliniar* desemneaza un algoritm generic de cautare în spatiul starilor. Alegerea unei metode sau alta de cautare în spatiul starilor este lasata la latitudinea celui care realizeaza implementarea.

Cea de a doua observatie pune în evidenta faptul ca ambele metode descrise în prima sectiune a articolului folosesc algoritmul c.m.m.p. neliniar. Indiferent de modul în care sunt estimate valorile initiale necesare procedurii ce implementeaza algoritmul c.m.m.p. neliniar, calitatea estimatiilor este dependenta, în primul rând, de calitatea acestui algoritm.

Cea de a treia observatie este ca algoritmul c.m.m.p. neliniar este utilizat, in cadrul bibliotecii ECONO si la estimarea modelelor STAR.

Cele trei observatii argumenteaza ideea ca atât alegerea unui algoritm c.m.m.p. neliniar, cât si implementarea sa trebuie tratate cu toata atentia.

La implementarea în biblioteca ECONO, algoritmul c.m.m.p. neliniar utilizat este Levenberg-Marquardt (LM). Algoritmul LM este considerat un standard pentru algoritmii c.m.m.p. neliniar. Argumentele pentru folosirea sa sunt implementarea relativ usoara, rezultatele excelente pe care le ofera si documentatia extinsa existenta.

Punctul de plecare în implementarea algoritmului LM a fost implementarea din biblioteca *Numerical Recipes in C (*NRC). Algoritmul este descris pe larg în [3], pag. 683-685. Sursele originale sunt listate în [3], pag. 685-687. În continuare se vor prezenta modificarile aduse implementarii din biblioteca NRC.

1. Tipul de data *float* utilizat în implementarea NRC a fost înlocuit cu tipul de data *double*. Modificarea a fost necesara pentru a asigura compatibilitatea cu restul functiilor din biblioteca ECONO.
2. Tipurile de date folosite pentru lucrul cu matrice au fost înlocuite cu tipurile de date specifice bibliotecii ECONO ([4]).
3. Prototipurile functiilor *mrqmin* si *mrqcoef* au fost modificate pentru a se putea calcula valoarea variabilei observate conform modelului ARMA ai carui parametrii trebuie estimati. Modificarile sunt prezentate în tabelul 2. Au fost facute, de asemenea, modificari în corpurile celor doua functii, dar si în corpurile functiilor *gaussj* si *covsrt* ce sunt apelate din functia *mrqmin*. Modificarile din functiile *mrqmin* si *mrqcoef*, necesare pentru calculul valorilor teoretice conform modelului ARMA, ar trebui sa fie evidente. Modificarile din functiile *gaussj* si *covsrt* decurg din tipurile de date utilizate si sunt minore.

Tabel 2. Modificarile aduse prototipurilor functiilor *mrqmin* si *mrqcoef*

Functia		**Semnificatia modificarilor**
Mrqmin		
Prototip initial	void mrqcof(float x[], float y[], float sig[], int ndata, float a[], int ia[],int ma, float **alpha, float beta[], float *chisq,void (*funcs)(float, float [], float *, float [], int))	Pentru semnificatia parametrilor vezi [3], pag. 683-687.

Prototip final	void mrqcof(Matrix* x, ColumnVector* resid, ColumnVector* y, ColumnVector sig, int ndata, ColumnVector a, int ia[],int p,int q, Matrix* alpha, ColumnVector* beta, double *chisq1, void (*funcs) (Matrix*, ColumnVector*, ColumnVector, double*, ColumnVector*, int,int,int));	Resid-vectorul termenului eroare din modelul ARMA dat de (2). Necesar pentru evaluarea componentei mobile a modelului. Transferul valorilor se face prin adresa. P, q-ordinul modelului ARMA. Inlocuieste ma-numarul de parametri ai modelului. Necesar pentru a face distinctia intre cele doua componente ale modelului ARMA.
Mrqcoef		
Prototip initial	void mrqmin(float x[], float y[], float sig[], int ndata, float a[], int ia[], int ma, float **covar, float **alpha, float *chisq, void (*funcs)(float, float [], float *, float [], int), float *alamda)	Pentru semnificatia parametrilor vezi [3]), pag. 683-687.
Prototip final	void mrqmin(Matrix* x, ColumnVector* y, ColumnVector sig, int ndata, ColumnVector* a, int ia[], int p,int q, Matrix *covar, Matrix *alpha, double *chisq1, void (*funcs)(Matrix*, ColumnVector*, ColumnVector, double*, ColumnVector*, int, int, int), double *alamda);	P,q-ordinul modelului ARMA. Vezi observatiile de mai sus.
Funcs		
	void funcs (float x[I], float a[], float *ymod, float dyda[], int ma))	Pentru semnificatia parametrilor vezi [3]), pag. 683-687.
	void funcs (Matrix*x, ColumnVector* resid, ColumnVector a, double*ymod, ColumnVector* dyda,int p,int q,int posx);	Resid-vectorul termenului eroare din modelul ARMA dat de (2). Necesar pentru evaluarea componentei mobile a modelului. Transferul valorilor se face prin adresa. P,q-ordinul modelului ARMA. Vezi observatiile de mai sus. Posx-indicele curent al variabilei calculate. Necesar pentru alegerea modului în care se face estimarea variabilei observate si reziduale.

5. Testarea metodei CalcARMA. Aspecte privind performantele numerice ale metodei. Comparatie cu alte pachete de programe.

Metoda care implementeaza estimarea parametrilor modelelor ARMA este denumita CalcARMA. Ea nu trebuie confundata cu metoda CalcCoef(), care exista înca din prima versiune a bibliotecii ECONO. Aceasta din urma calculeaza parametrii modelelor AR pornind de la functia de autocorelatie a acestora.

Testarea algoritmului dupa modificarile aduse a fost efectuata în urmatorul fel:

- Dupa modificarea tipurilor de date algoritmul a fost testat pe date simulate. În simulare s-au utilizat modelele: liniar, exponential si logaritmic.

- Pe lânga datele simulate au fost folosite si seturi de date de test. Acestea pot fi gasite la adresa http://www.netec.mcc.ac.uk clasificate în functie de dificultatea lor. Aceste date sunt utilizate pentru evaluarea programelor de analiza statistica si econometrica de catre potentialii beneficiari, reviste de specialitate etc. Testarea algoritmului a fost facuta cu succes pentru seturi de date clasificate, având niveluri de dificultate *redus*, *mediu* si *dificil*. Pentru cele clasificate ca *foarte dificile* au aparut probleme din punctul de vedere al rapiditatii convergentei si/sau calitatii estimatorilor. Aceasta nu este o surpriza, algoritmul având o arie de aplicabilitate limitata si cunoscuta. În plus, în contextul seriilor de timp este putin probabil sa întâlnesti astfel de ecuatii complicate.
- Dupa modificarea functiilor originale, astfel încât sa permita calcularea modelelor ARMA, algoritmul a fost testat pe date simulate si reale. Din pacate, nu am gasit serii de timp de test pentru a compara rezultatele obtinute cu cele corecte. Datele simulate utilizate în testare au fost:
- trei modele ARMA (1,0) cu parametrul autoregresiv –0,5; 0,3; 0,95. Pentru fiecare model s-au simulat serii de date cu lungimile 40, 120, 300.
- Patru modele ARMA cu parametrii (ϕ=-0,5 θ=0,5) ; (ϕ=-0,3 θ=0,7); (ϕ=-0,94 θ=0,5); (ϕ=-0,9 θ=0,95). Pentru fiecare model au fost generate serii de timp cu lungimi de 40, 120, 300.

Rezultatele obtinute conduc la urmatoarele concluzii:

a) Rezultatele sunt foarte bune atunci când parametrii modelului ARMA nu se apropie în valoare absoluta de unu. Rezolvarea poate fi prin: utilizarea metodei backforecasting pentru obtinerea unor estimari initiale ale parametrilor si utilizarea lor ca valori de start pentru metoda c.m.m.p. conditionate. Experienta numerica de pâna acum confirma eficienta acestei metode.

b) Calitatea estimatorilor obtinuti prin cele doua metode creste mai rapid în cazul metodei c.m.m.p. conditionate. Cu alte cuvinte, comportamentul asimptotic al estimatorilor obtinuti prin aceasta metoda este mai bun decât în cazul metodei backforecasting.

c) Viteza cu care converge algoritmul este afectata doar în cazul în care parametrii modelului se apropie în valoare absoluta de unu. Pentru celelalte modele testate convergenta se atinge în maxim 8 iteratii.

d) Comparatia cu alte pachete de programe s-a facut folosind modulul *Time Series* al pachetului Statistica 5.01 si programul EconometricViews. Performantele obtinute de catre ECONO sunt comparabile, din toate punctele de vedere, cu cele obtinute prin utilizarea EconometricViews. Comparativ cu Statistica, ECONO este depasit în cazul seriilor de timp de dimensiuni mici si în cazul în care parametrii modelului se apropie în modul de unu. Aceasta se explica prin faptul ca, dupa toate aparentele, Statistica maximizeaza functia de verosimilitate data de (4) si nu cea redusa data de (5).

Bibliografie

1. Terence C. Mills, Time series techniques for economists, Cambridge University Press, 1990
2. Eugen Pecican, Econometrie, Editura All, Bucuresti, 1995
3. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Numerical Recipes in C, Cambridge University Press, 1995
4. R.B. Davies, Documentation for newmat09, a matrix library în C++, http://webnz.com/robert/ ,1998
5. Adrian Hospodar, Documentatie pentru biblioteca ECONO, Bucuresti, 1999