

Модели динамического стохастического общего экономического равновесия (ДСОЭР). Ошибки численных методов¹

Иващенко С.М.

Приложение

В файле `dyn_first_order_solver.m` нужно вставить следующий кусок кода, чтобы можно было использовать различный порядок подачи матриц в алгоритм QZ-декомпозиции.

```
if isfield(DynareOptions, 'always_BK_simple')
    if DynareOptions.always_BK_simple==1
        [temp_AA,temp_BB,temp_Q,temp_Z] = qz(E,D, 'real');

        [ei,ii]=sort(-abs(ordeig(temp_AA,temp_BB)));
        ii2=(ii~=0);
        ii2(ii(1:nsfwrd))=0;
        [ss,tt,temp_QS,w] =
ordqz(temp_AA,temp_BB,temp_Q,temp_Z,ii2);
        sdim=nd-nsfwrd;
        nba=nsfwrd;
    end
    if DynareOptions.always_BK_simple==-1
        [temp_BB,temp_AA,temp_Q,temp_Z] = qz(D,E, 'real');

        [ei,ii]=sort(-abs(ordeig(temp_BB,temp_AA)));
        ii2=(ii~=0);
        ii2(ii(end+1-nsfwrd:end))=0;
        [tt,ss,temp_QS,w] =
ordqz(temp_BB,temp_AA,temp_Q,temp_Z,ii2);
```

¹ Исследование выполнено при частичной финансовой поддержке РФФИ, проект «Структурные изменения в экономике России: роль человеческого капитала и инвестиций» № 18-010-01185.

Иващенко Сергей Михайлович – к.э.н., с.н.с. Института проблем региональной экономики РАН; Центра перспективного финансового планирования, макроэкономического анализа и статистики финансов, Научно-исследовательского финансового института Минфина РФ; факультет экономики СПбГУ. E-mail: sergey.ivashchenko.ru@gmail.com

Статья поступила: 24.07.2018/Статья принята: 20.09.2018.

```

        sdim=nd-nsfwrд;
        nba=nsfwrд;
    end
end

```

Данный кусок кода следует вставить после того как dynare проверит qz-алгоритм на ошибки (см. следующий кусок кода dynare).

```

if info1
    if info1 == -30
        % one eigenvalue is close to 0/0
        info(1) = 7;
    else
        info(1) = 2;
        info(2) = info1;
        info(3) = size(E,2);
    end
    return
end

nba = nd-sdim;

```

Для того чтобы провести балансировку матриц, нужно вставить следующий кусок кода.

```

if isfield(DynareOptions, 'balance_qz')
    if DynareOptions.balance_qz==1
        [Dl, Dr, iter] = baleig(E,D,5);
        E_init=E;E=diag(1./Dl)*E*diag(Dr);
        D_init=D;D=diag(1./Dl)*D*diag(Dr);
    end
end

```

Этот кусок кода надо вставить перед расчетом QZ-декомпозиции встроенной в dynare (см. следующий кусок кода dynare).

```

[err, ss, tt, w, sdim, dr.eigval, info1] = mjdgges(E, D, DynareOptions.qz_criterium, DynareOptions.qz_zero_threshold);
mexErrCheck('mjdgges', err);

```

Следующий кусок кода надо вставить для учета балансировки матриц.

```

if isfield(DynareOptions, 'balance_qz')
    if DynareOptions.balance_qz==1
        gx=diag(Dr(indx_explosive_root))*gx*diag(1./Dr(indx_stable_root));
        hx=diag(Dr(indx_stable_root))*hx*diag(1./Dr(indx_stable_root));
        ghx = [hx(k1, :); gx(k2(nboth+1:end), :)];
    end
end

```

Кусок кода вставляется после основных расчетов `dynare` (перед расчетом для статических переменных), т.е. следует вставить после следующего куска кода `dynare`:

```
opts.TRANSA = true;
hx1 = linsolve(tt(indx_stable_root,
indx_stable_root), Z11, opts)';
opts.UT = false; % needed by Octave 4.0.0
opts.TRANSA = false; % needed by Octave 4.0.0
hx2 = linsolve(Z11, ss(indx_stable_root,
indx_stable_root)', opts)';
hx = hx1*hx2;
ghx = [hx(k1, :); gx(k2(nboth+1:end), :)];
```

Также необходимо ввести несколько команд в файл модели (`mod`-файл), чтобы рассчитать меры точности расчета линейной аппроксимации. Необходимо симулировать данные-выборку, задать начальную точку расчетов, рассчитать значения функции правдоподобия и моменты с двумя вариантами порядка подачи матриц в `QZ`-декомпозицию, с балансировкой матриц и без балансировки. Пример куска кода для модели [Chari et al., 2007], который следует вставить после остального кода модели:

```
@#if 1==1
stoch_simul(nograph, noprint, order=1, drop=200, periods=300, replic=1)
y c k x l w z g tau_l tau_x log_labor_wedge
log_investment_wedge log_efficiency_wedge ;

n=120; for
i=1:size(M_.endo_names,1); eval([M_.endo_names(i,:), '=', M_.endo_names(i,:), '(end-n+1:end,1);']); end;
save temp y c k x l w z g tau_l tau_x log_labor_wedge
log_investment_wedge log_efficiency_wedge ;

@#endif

varobs y c k x l w z g tau_l tau_x log_labor_wedge
log_investment_wedge log_efficiency_wedge ;
estimated_params;
stderr y , 1.0e-4, 0, 1, normal_pdf, 0, 1.0e-4;
stderr c , 1.0e-4, 0, 1, normal_pdf, 0, 1.0e-4;
stderr k , 1.0e-4, 0, 1, normal_pdf, 0, 1.0e-4;
stderr x , 1.0e-4, 0, 1, normal_pdf, 0, 1.0e-4;
stderr l , 1.0e-4, 0, 1, normal_pdf, 0, 1.0e-4;
stderr w , 1.0e-4, 0, 1, normal_pdf, 0, 1.0e-4;
stderr z , 1.0e-4, 0, 1, normal_pdf, 0, 1.0e-4;
stderr g , 1.0e-4, 0, 1, normal_pdf, 0, 1.0e-4;
stderr tau_l , 1.0e-4, 0, 1, normal_pdf, 0, 1.0e-4;
stderr tau_x , 1.0e-4, 0, 1, normal_pdf, 0, 1.0e-4;
```

```
stderr log_labor_wedge      , 1.0e-4,0,1, normal_pdf,0,1.0e-4;
stderr log_investment_wedge, 1.0e-4,0,1, normal_pdf,0,1.0e-4;
stderr log_efficiency_wedge, 1.0e-4,0,1, normal_pdf,0,1.0e-4;

end;

xparam1=ones(13,1)*1.0e-4;hh=eye(13);save temp_mode xparam1 hh;

@if 1==1
options_.always_BK_simple=1;disp('test 1');
estimation(datafile=temp,mode_compute=0,first_obs=1, presam-
ple=0,lik_init=1,kalman_tol =1.0e-20,mcmc_jumping_covariance
=identity_matrix,mode_file=temp_mode
,prefilter=0,mh_replic=0,mh_nblocks=2,mh_jscale=0.20,mh_drop=0.2,
nograph, nodiagnostic, cova_compute=1);
stoch_simul(nograph, noprint,order=1, replic=1,periods=0)
;oo_1=oo_;
options_.always_BK_simple=-1;disp('test -1');
estimation(datafile=temp,mode_compute=0,first_obs=1, presam-
ple=0,lik_init=1,kalman_tol =1.0e-20,mcmc_jumping_covariance
=identity_matrix,mode_file=temp_mode
,prefilter=0,mh_replic=0,mh_nblocks=2,mh_jscale=0.20,mh_drop=0.2,
nograph, nodiagnostic, cova_compute=1);
stoch_simul(nograph, noprint,order=1, replic=1,periods=0)
;oo_2=oo_;

options_.balance_qz=1;
options_.always_BK_simple=1;disp('test 1,balance');
estimation(datafile=temp,mode_compute=0,first_obs=1, presam-
ple=0,lik_init=1,kalman_tol =1.0e-20,mcmc_jumping_covariance
=identity_matrix,mode_file=temp_mode
,prefilter=0,mh_replic=0,mh_nblocks=2,mh_jscale=0.20,mh_drop=0.2,
nograph, nodiagnostic, cova_compute=1);
stoch_simul(nograph, noprint,order=1, replic=1,periods=0)
;oo_3=oo_;
options_.always_BK_simple=-1;disp('test -1,balance');
estimation(datafile=temp,mode_compute=0,first_obs=1, presam-
ple=0,lik_init=1,kalman_tol =1.0e-20,mcmc_jumping_covariance
=identity_matrix,mode_file=temp_mode
,prefilter=0,mh_replic=0,mh_nblocks=2,mh_jscale=0.20,mh_drop=0.2,
nograph, nodiagnostic, cova_compute=1);
stoch_simul(nograph, noprint,order=1, replic=1,periods=0)
;oo_4=oo_;
    @#endif
```