```
function [N,G] = GCRDr(P,tol)
%
% The function [N,G] = GCRDr(P,tol)
% computes a compact GCRD factorization P(s)=N(s)*G(s) where
% - N(s) has a polynomial left inverse
% - G(s) has full normal row rank r and has the
%   same Smith zeros and right minimal indices as P(s)
% - all matrices are polynomial and stored as a 3D array.
% - tol is a tolerance used in the rank decisions
%
mnd=size(P);n=mnd(2);d=mnd(3)-1;
% Construct the initial generalized state space model
Qin=eye(d*n,d*n);Zin=eye(n+d*n,n+d*n);
Ein=Zin(n+1:n+d*n,:);
Ain=Zin(1:d*n,:);
Bin=-Zin(:,n*d+1:n*d+n);
Cin=P(:,:,1);for i=1:d, Cin=[P(:,:,i+1) Cin];end
[C0,Q0,Z0,E0,A0,r4] = Initialize(P,tol);
[Q,Z,E,A,mcur,ncur,s,t] = Staircase(E0,A0,Q0,Z0,r4,tol);
% Select subpencil and treat it separately
mn=size(A);mm=mn(1);nn=mn(2);
Esub=E(mcur+1:mm,ncur+1:nn);
Asub=A(mcur+1:mm,ncur+1:nn);
Csub=C0(:,ncur+1:nn);
mup=mm-mcur;nup=nn-ncur;r=nup-mup;
Qsub=eye(mup,mup);
Zsub=eye(nup,nup);
[Qup,Zup,Eup,Aup,Cup,Ahat,s,t,k]=Embed(Esub,Asub,Csub,Qsub,Zsub,tol);
% Construct the left factor N(s)
Bup=zeros(mup,r);Bup(mup+1:nup,:)=eye(r,r);
EN=Eup;EN(nup,nup)=0;AN=[Aup;Ahat];
Ainv=inv(AN);AEN=Ainv*EN;AB=Ainv*Bup;
N(:,:,1)=Cup*AB;
for i=1:d, AB=AEN*AB;N(:,:,i+1)=Cup*AB; end
N=Trim(N,tol);
% Construct the feedback F and then G(s)
F=Zin(n*d+1:n*d+r,:);
F=F-Ahat*Zup'*Z(:,n*d+n-nup+1:n*d+n)';
for i=1:d,G(:,:,d+1-i)=-F(:,n*i+1:n*(i+1));end
G(:,:,d+1)=-F(:,1:n);G(:,:,1)=G(:,:,1)+eye(r,n);
G=Trim(G,tol);
return
```