Stochastic Simulations with the model PANTA RHEI

by Christian Lutz
GWS mbH

Presentation at the 9th INFORUM World Conference in Gerzensee, 9-16th September 2001

1. Why stochastic simulations?

- Four sources of errors in multi-equation simulation models:
  - equation contains implicit additive error term,
  - estimated values of coefficients are themselves random variables,
  - exogenous variables have to be forecasted, and those forecasts may contain errors
  - equation is misspecified (functional form does not represent the real world).

- Limitation to the first source of error:

$Y_t = \quad + ß X_t + \quad_t$ with $\quad_t$
N(0, ^{2})

2. Implementation in PANTA RHEI

- For 1173 estimated equations:
  - calculate standard error of estimation:

```
ti food
lim 83 96
r cpvr01 = YVH, ZEIT
fex CPVR01_S = @sum(@sq( resid )) / 14
```

  - generate random variables and transform to $N(0, S^2)$:

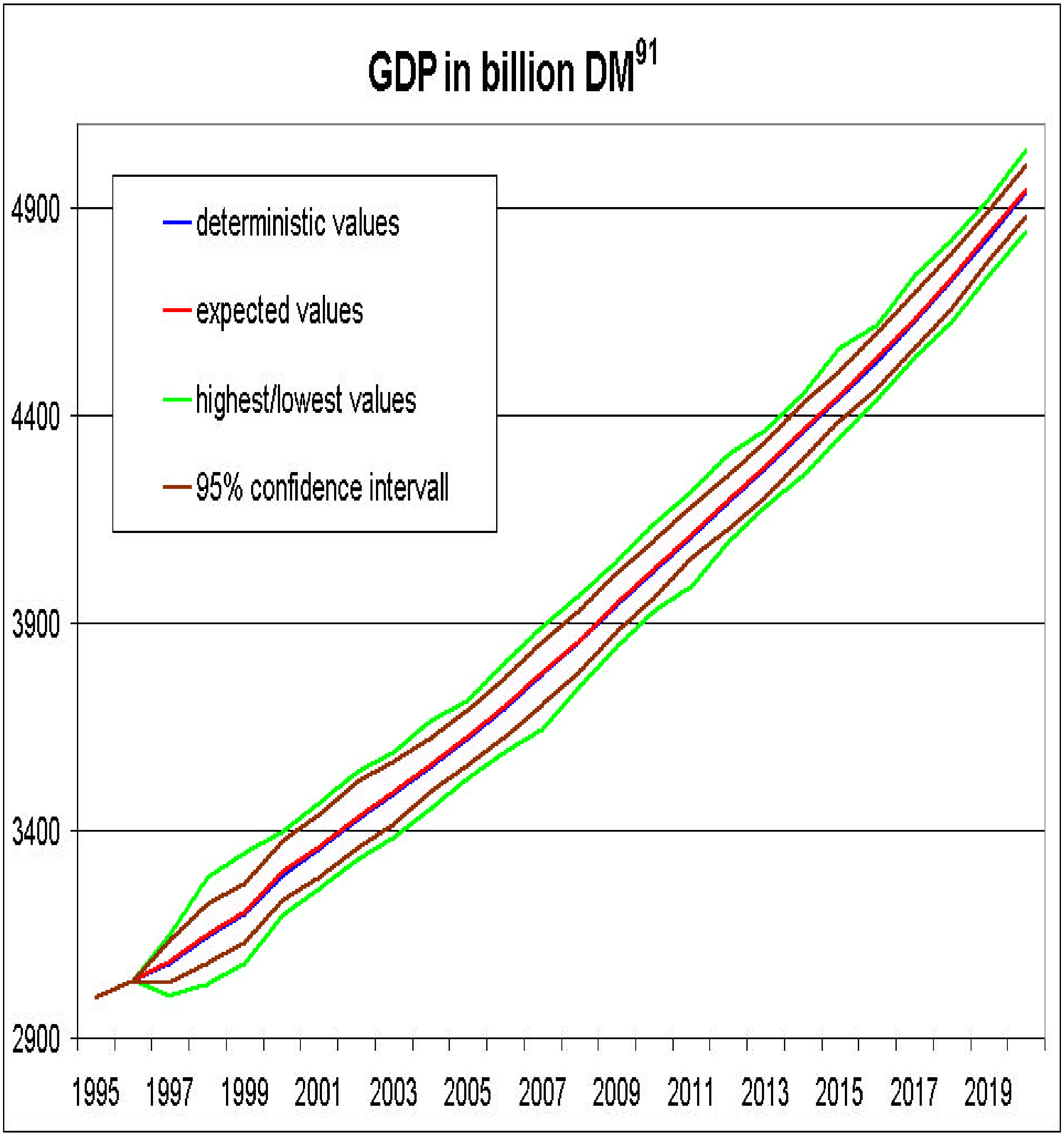
```
if (iter == 1 && t > 1996)
   CPVR01_S[t] = ndev( CPVR01_S[1996] );
```

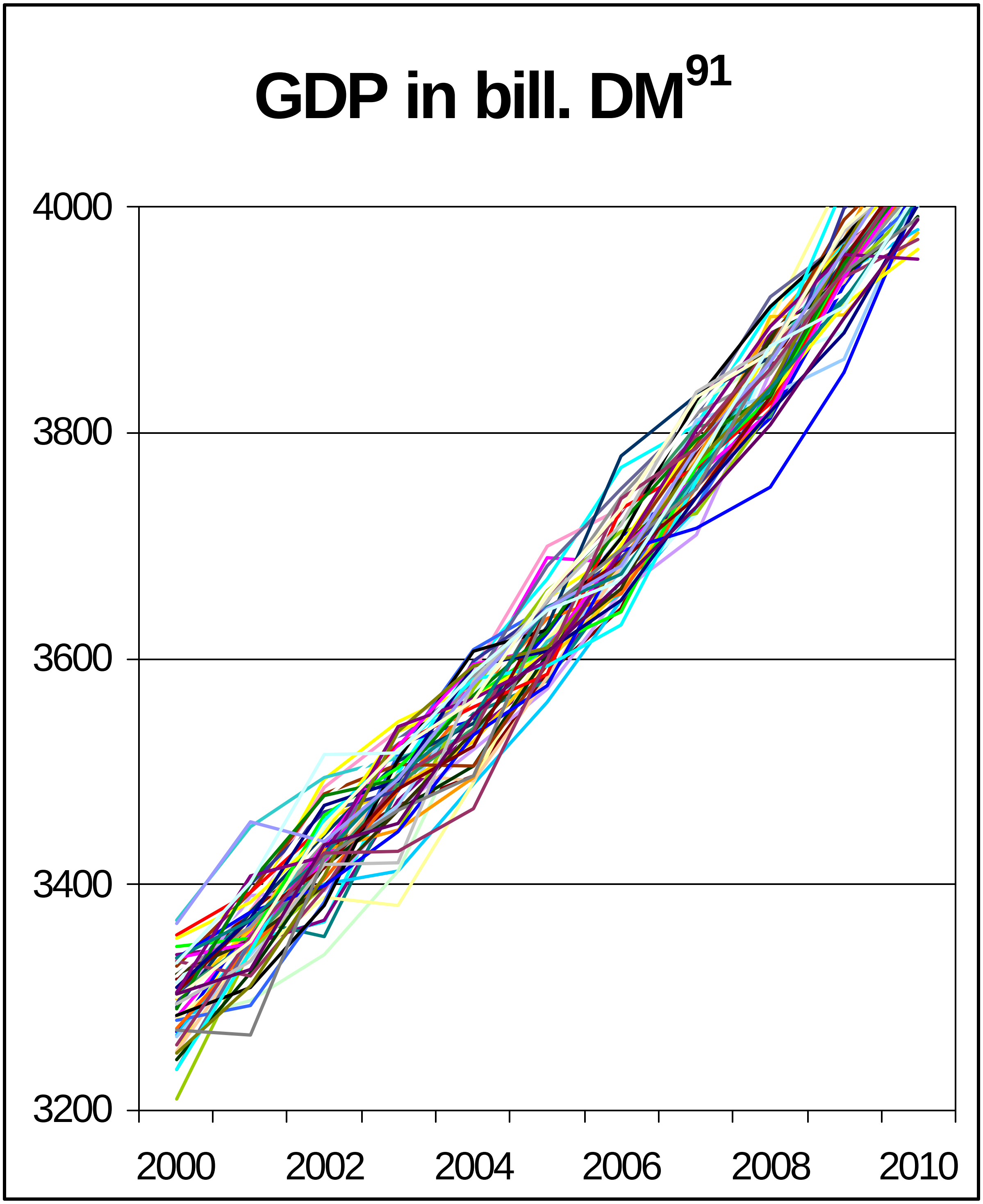
  - add random variables to the equations (assuming covariances between coefficients to be zero):

```
cpvrEQN.rhostart = cpvr.LastData();
depend = 172.626191*intercept + 0.104478*YVH[t]
   + -1.871435*ZEIT[t];
if (t > 1996) depend += CPVR01_S[t];
cpvr[1] = cpvrEQN.rhoadj(depend, cpvr[1], 1);
```

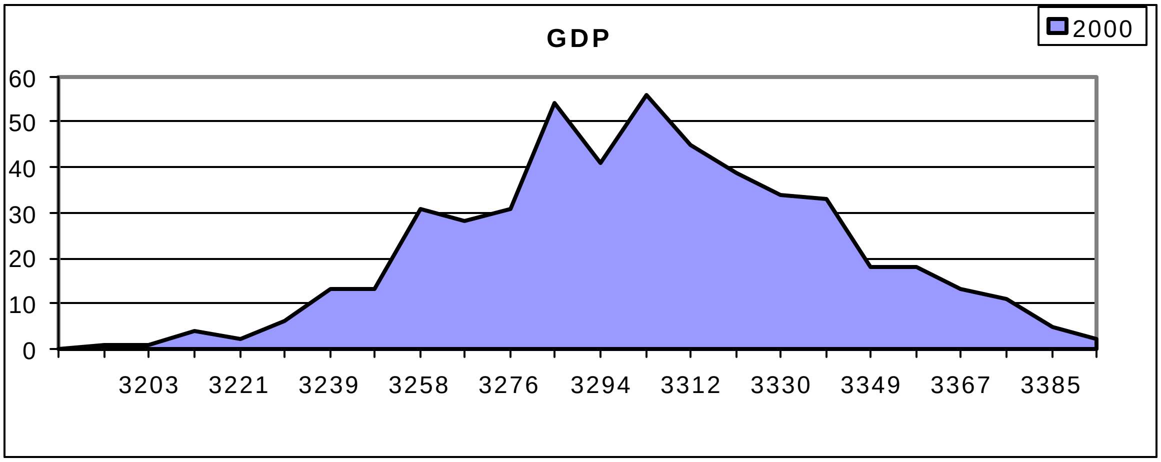
3. Results of 500 simulations

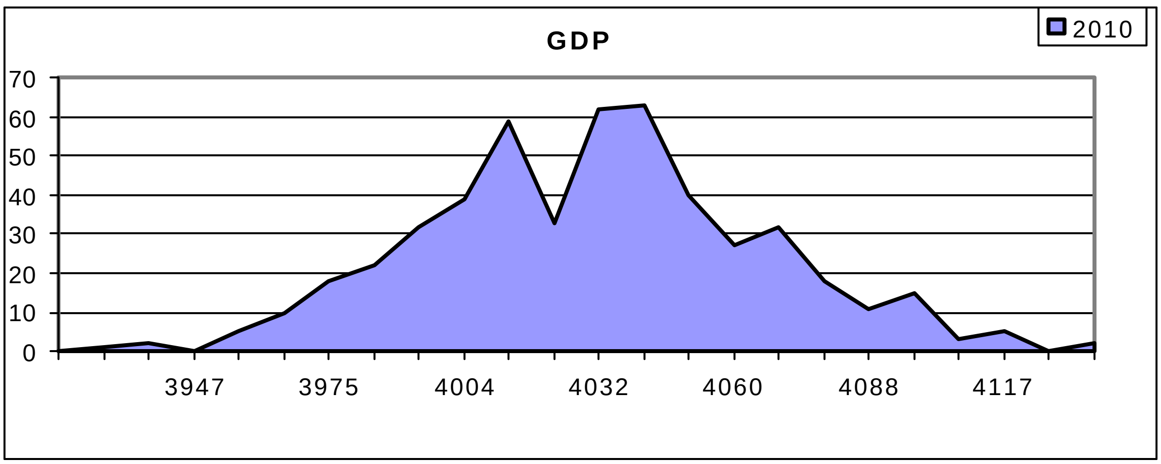
(10 runs did not converge)

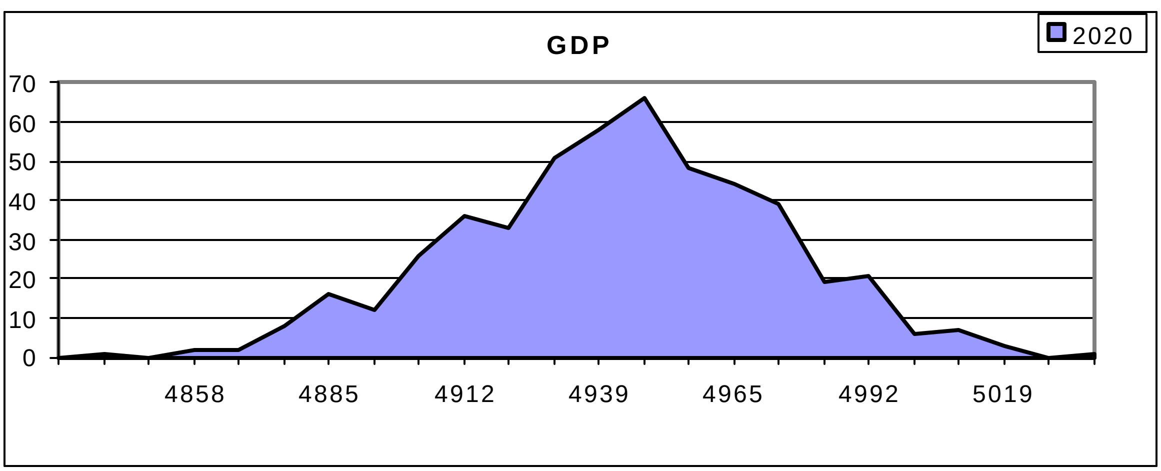



GDP in bill. DM91
4000
3800
3600
3400
3200
2000
2002
2004
2006
2008
2010

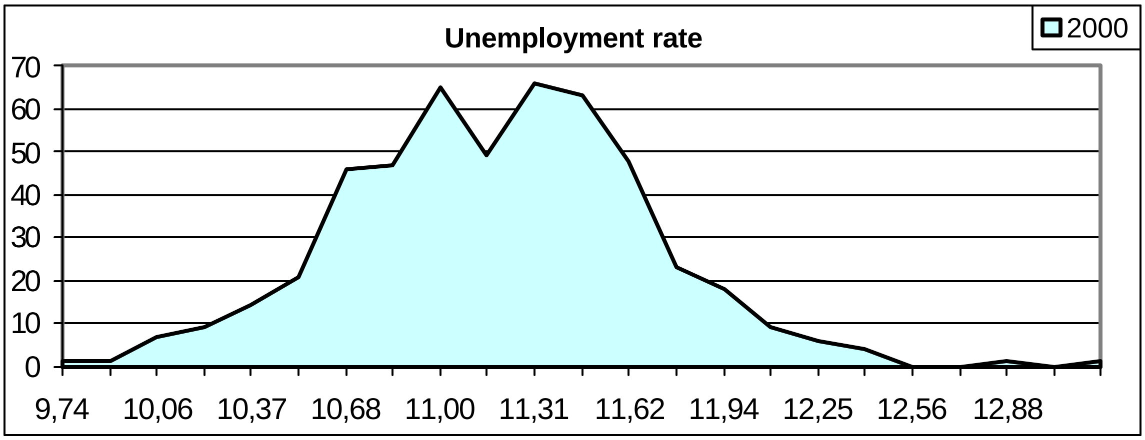
Histograms of GDP in bill. DM^{91}

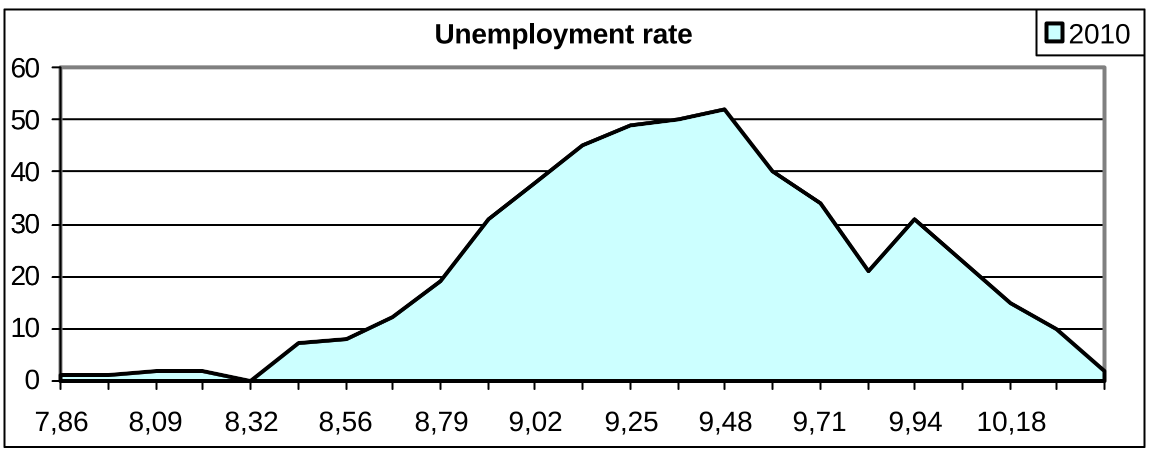






Histograms of unemployment rate in %





4. Conclusion

- Similar results for other variables
- Additive error terms are not important
  => deterministic simulation is enough!
- Dynamic interdependencies small
  (exception: public consumption,
  small influence of capital stocks
  in the model version)
- Focus on other sources of error:
  - errors in exogenous variables
    => sensitivity analysis
  - misspecified equations
    => historical (ex post) simulations
    (simulation errors)
    => overall sensitivity analysis
    (ex ante simulations, use of the model)

Appendix

A1: Random numbers between 0 and 1

```
#define IM_s 2147483647
#define IA_s 16807
#define AM_s (1.0/IM_s)
#define IQ_s 127773
#define IR_s 2836
#define NTAB_s 32
#define NDIV_s (1+(IM_s-1)/NTAB_s)
#define EPS_s 1.2e-7
#define RNMX_s (1.0-EPS_s)

// generate random numbers

float ran1 (long *idum)

{       int j;
        long k;
        static long iy = 0;
        static long iv [NTAB_s];
        float temp;

        if (*idum <= 0 ||! iy){

          if (-(*idum) < 1)  *idum = 1;
          else *idum = -(*idum);

          for (j=NTAB_s + 7; j >= 0; j--){
                  k=(*idum)/IQ_s;
                  *idum = IA_s*(*idum - k*IQ_s ) - IR_s*k;
                  if (*idum<0) *idum += IM_s;
                  if (j<NTAB_s) iv[j] = *idum;
      }
             iy = iv [0];
        }

        k = (*idum) / IQ_s;
      *idum = IA_s*(*idum-k*IQ_s) - IR_s*k;

        if (*idum < 0)  *idum += IM_s;

        j = iy / NDIV_s;
        iy = iv [j];
        iv [j] = *idum;

        if ((temp = AM_s*iy) > RNMX_s) return RNMX_s;
        else return temp;
}
```

A 2: Change random numbers into normal distribution

```
// Normal distribution

float nver (float *var, long *idum)
{
      static int iset=0;
      static float gset;
      float fac, rsq, v1, v2;

      if (*idum <0) iset = 0;

      if (iset == 0) {

        do {
              v1=2.0*ran1(idum)-1.0;
              v2=2.0*ran1(idum)-1.0;
              rsq=v1*v1+v2*v2;
        } while (rsq >= 1.0 || rsq== 0.0);

          fac = sqrt(-2.0*log(rsq)/rsq);


          gset = v1*fac;
          iset = 1;
          return v2*fac*sqrt (*var);

      } else {            /* (iset != 0) */

          iset = 0;
          return gset*sqrt (*var);
      }

}
```