



RICERCA DI SISTEMA ELETTRICO

Development of Monte Carlo Algorithms for Eigenvalue Calculations

Ken William Burn

Report RdS/2012/002

DEVELOPMENT OF MONTE CARLO ALGORITHMS FOR EIGENVALUE CALCULATIONS

Ken William Burn (ENEA)

Settembre 2012

Report Ricerca di Sistema Elettrico Accordo di Programma Ministero dello Sviluppo Economico - ENEA Area: Governo, gestione e sviluppo del sistema elettrico nazionale Progetto: Nuovo nucleare da fissione: collaborazioni internazionali e sviluppo competenze in materia nucleare

Responsabile del Progetto: Paride Meloni, ENEA

ENEN Ricerca Sistema Elettrico				Sigla di identificazione		Distrib.	Pag.	di
			trico	NNFISS – LP	2 – 083	L	1	10
Γit	olo Development o Calculations	f Monte Ca	arlo Algo	orithms for Eige	nvalue			
De	scrittori Tipologia del documento: Collocazione contrattuale Argomenti trattati:	Rapporto : Accordo nucleare Metodi d	Tecnico di progra da fissic li calcolo	amma ENEA-MS one" neutronico. Meto	E: tema d	i ricerca	"Nuovo Riduzio) ne
s a	mmario a first stage in the developr	ment of new a	algorithms	to reduce the varia	nce in eigen	value calc	ulations	with
ne nit oll ptii	source-iteration approach, a v-value to ensure a supply y has not yet been inserted. owing this, the developmen mum population within eac erhistory); secondly a searc	a "superhisto of children at .t will be artic ch cycle (or .h for the trad	ry" approad t each norr ulated in tw superhisto e-off betwe	ch has been introdu malization in the cas wo steps – firstly a s ory) and at the new een cycle size and r	ced into MC se that keff c search for th kt normaliza sumber of no	NP5. An a differs appr e trade-off tion point prmalizatio	ndjustme reciably f betwee (i.e. en n points.	from from n an nd of
he init oll opti up	source-iteration approach, a v-value to ensure a supply y has not yet been inserted. owing this, the developmen mum population within eac erhistory); secondly a searc	a "superhisto of children af t will be artic ch cycle (or th for the trad	ry" approad t each norr ulated in tw superhisto le-off betwo	ch has been introdu malization in the cas wo steps – firstly a s ory) and at the new een cycle size and r	ced into MC se that keff c search for th at normaliza number of no	NP5. An a differs appr e trade-off tion point prmalizatio	ndjustme reciably f betwee (i.e. en n points	int of from in an ind of
ne nit oll pti up	source-iteration approach, a v-value to ensure a supply y has not yet been inserted. owing this, the developmen mum population within eac erhistory); secondly a searc te Autor	a "superhisto of children af t will be artic ch cycle (or th for the trad	ry" approa t each norr ulated in tw superhisto le-off betwo urn In caric	ch has been introdu malization in the cas wo steps – firstly a s ory) and at the nex een cycle size and r	ced into MC se that keff c search for th kt normaliza number of no	NP5. An a differs appr e trade-off tion point prmalizatio	ndjustme reciably betwee (i.e. en n points	ent of from en an ad of
	source-iteration approach, a v-value to ensure a supply y has not yet been inserted. owing this, the developmen mum population within eac erhistory); secondly a searc te Autor	a "superhisto of children af t will be artic ch cycle (or th for the trad	ry" approad t each norr ulated in tw superhisto le-off betwo le-off betwo urn In caric NOME	ch has been introdu malization in the cas wo steps – firstly a s ory) and at the nex een cycle size and r	ced into MC se that keff c search for th kt normaliza number of no	NP5. An a differs appr e trade-off tion point prmalizatio	ndjustme reciably betwee (i.e. en n points	ent of from en ar ad of
	source-iteration approach, a v-value to ensure a supply y has not yet been inserted. owing this, the developmen mum population within eac erhistory); secondly a searc te Autor	a "superhisto of children af t will be artic ch cycle (or th for the trad	ry" approad t each norr ulated in tw superhisto le-off betwo le-off betwo le nome Firma	ch has been introdu malization in the cas wo steps – firstly a s ory) and at the nex een cycle size and r	ced into MC se that keff c search for th kt normaliza number of no	NP5. An a differs appr e trade-off tion point prmalizatio	ndjustme reciably betwee (i.e. en n points	ent of from en ar ad of
	source-iteration approach, a v-value to ensure a supply y has not yet been inserted. owing this, the developmen mum population within eac erhistory); secondly a searc te Autor	a "superhisto of children af t will be artic ch cycle (or th for the trad	ry" approad t each norr ulated in tw superhisto le-off betwo le-off betwo le nome Firma NOME	ch has been introdu malization in the cas wo steps – firstly a s ory) and at the nex een cycle size and r	ced into MC se that keff c search for th kt normaliza number of no	NP5. An a differs appr e trade-off tion point prmalizatio	ndjustme reciably f betwee (i.e. en n points	ent o from en ar ad o
	source-iteration approach, a v-value to ensure a supply y has not yet been inserted. owing this, the developmen mum population within eac erhistory); secondly a searc te Auto ia n.	a "superhisto of children af t will be artic ch cycle (or th for the trad	ry" approad t each norr ulated in tw superhisto le-off betwo le-off betwo le-off betwo Firma NOME FIRMA NOME FIRMA	o a:	ced into MC se that keff c search for th kt normaliza number of no	NP5. An a differs appr e trade-off tion point prmalizatio	djustme reciably betwee (i.e. en n points	ent of from en ar ad of
he he init option option op 2 	source-iteration approach, a v-value to ensure a supply y has not yet been inserted. owing this, the developmen mum population within eac erhistory); secondly a searc fie Autor ia n.	a "superhisto of children af t will be artic ch cycle (or th for the trad	ry" approad t each norr ulated in tw superhisto le-off betwo le-off betwo le-off betwo Firma NOME Firma NOME	ch has been introdu malization in the cas wo steps – firstly a s bry) and at the nex een cycle size and r o a: K. W. Burn	F. Pado	NP5. An a differs appr e trade-off tion point prmalizatio	P. Melo	ent of from en an ad of



Index

Index	2
1. Introduction	3
2. Discussion of Approach	3
3. Superhistories	3
4. Current State of Project	4
References	4
Appendix: Superhistory Patch to MCNP5-1.4	5



ENEN Ricerca Sistema Elettrico

1. Introduction

For many years optimization of fixed source Monte Carlo radiation transport calculations has been under development at ENEA [1]. Recently the focus has been redirected from fixed source to eigenvalue calculations employing the source iteration approach.

At first sight the developments at ENEA, named the "DSA" (Direct Statistical Approach to Variance Reduction Optimization), look to be well suited to deal with the source iteration approach as:

- a) they explicitly take into account correlations between progeny coming from the same progenitor, be the event an artificial bifurcation (viz. splitting) or a natural bifurcation [e.g. (n,xn), fission].
- b) They search for the optimum trade-off between greater splitting that decreases the second moment and increases the time (and thereby allows less source particle histories to be run in a given time) and greater Russian roulette that increases the second moment and decreases the time (and thereby allows more source particle histories to be run in a given time).

In the source iteration method, correlations between fission daughters contribute to underestimating the standard error, which is calculated under the assumption that fission generations are independent. Furthermore the balance between the number of fission neutrons per generation and the total number of generations is decided empirically. Both these issues look susceptible to an approach such as the DSA.

Furthermore the question of calculating a neutronic/photonic response in a spatially limited domain in a source-iteration calculation remains an unsolved problem.

2. Discussion of Approach

Initially an attempt was made to treat the source-iteration problem as a fixed source problem by firstly assuming that the fundamental mode of fission sites has been reached and secondly sampling fission chains starting from this source, normalizing the result from each source history by both the length of the fission chain. Unfortunately calculating responses in this way gives a first moment that tends to decrease dramatically as the number of cycles increases. This is because more successful fission neutrons tend to make longer fission chains, and because of the normalization by the length of each fission chain, tend to count less than less successful fission neutrons with shorter fission chains (due to absorption, leakage, etc.).

An approach nearer the source-iteration algorithm looked to be indicated. As a first step the number of fission site renormalizations was reduced from "at every fission generation" to "at every L fission generations", i.e. the "superhistory" approach.

3. Superhistories

Reducing the number of fission site renormalizations was first proposed in [2] to reduce the bias in the estimated variance. Here it is employed instead as a first step in an approach to improve variance reduction.

The DSA is not a stand-alone code but employs the well-known general-purpose radiation transport code, MCNP [3], as a vehicle. Thus any code is distributed as a "patch" to MCNP that could also be modified to patch another code.

As a first step, a superhistory patch has been inserted into MCNP5 (ver. 1.4). This patch is shown in the Appendix. An adjustment of the v-value to ensure a supply of children at each



ENEN Ricerca Sistema Elettrico

normalization in the case that k_{eff} differs appreciably from unity, as made in [2], has not yet been inserted.

4. Current State of Project

The superhistory patch runs and has been tested. The current version of the DSA patch for fixed source problems was updated between 2010 and 2011 to be compatible with MCNP5 (ver. 1.3) and in particular to run with MPI. Currently the DSA patch is being updated to MCNP5 (ver. 1.4). (An important advantage of ver. 1.4 over ver. 1.3 is the presence of the Shannon entropy diagnostics.) It will then be combined with the superhistory patch and further developed to calculate spatially limited neutronic responses in reactor cores (e.g. in-core neutron detectors) and neutron detector responses near fissile storage arrays. The development will be articulated in two steps – firstly a search for the trade-off between an optimum population within each cycle (or superhistory) and at the next normalization point (i.e. end of superhistory); secondly a search for the trade-off between cycle size and number of normalization points.

References

[1] K.W. Burn "Optimizing Monte Carlo to Multiple Responses: the Direct Statistical Approach, 10 Years On", Nucl. Technol. 175, 138 (2011).

[2] R.J. Brissenden, A.R. Garlick "Biases in the Estimation of K_{eff} and Its Error by Monte Carlo Methods", Ann. Nucl. Energy 13-2, 63 (1986).

[3] X-5 Monte Carlo Team "MCNP – A General Monte Carlo N-Particle Transport Code, Version 5", LA-UR-03-1987 (2003)



Appendix: Superhistory Patch to MCNP5-1.4

```
*ident,May.12<<<<<
*/
*/
    Version 1:
*/
   This version inserts a superhistory (R. Brissenden) option into kcode calculations.

idum(3) (input) is the no. of fission events between renormalizations - = 0 if

it is a normal calculation (no superhistories).

idum(4) is a count (down) of the number of fission events since the last renormalization.
*/
*/
*/
*/
*open,mcnp_global.F90
*/
*/
*/
               VDAC
*/
nsl(:,:), & != Summary information for surface source file.
   &
               != Counts of scores beyond the last bin.
   &
    ntbb(:,:)
*/
*/
*/
*open,varcom.F90
*d,varcom.48
          + 1*(50) + 1*(2*99) + 1*(100) + 3*(1) + 1*(3) + 1*(1)
*i,varcom.177
 integer :: nsz
                           != Count of starter + in-superhistory fission source points
                            (in master) or just in-superhistory fission source points
                           ! (in slaves).
*d,varcom.215,varcom.222
   ! non-swept integer variables.
 common /varcm/ &
   & hsrc auto mesh,
                      hsrc init mesh,
                                       hsrc nmsh,
                                                   hsrc ixyz,&
                                                   kcsf,
   & irmc, ist, ist0, ixak, ixak0, jrad, kc8,
& kct, kcy, knod, ksdef, lost, monod, nbov,
& nesm, netb, nfer, notal, notrn, npnm, nppm,
                                            kcz,
                                                        £
                                             ndmp,
                                                  nerr,
                                                        &
                                             npum,
                                                   nsa,
                                                        æ
   & nsa0, nskk, nsom, nss, nss0, nsz,
& nwse, nwsg, nwst, nwws, mvarcm
                                       nssi,
                                            ntprt, nwer, nwsb,
                                                             &
*/
*/
*/
*open,setdas.F90
* /
*i.setdas.502
 allocate( lfclsv( 1:mxa1+2 ) )
*/
*/
*i,setdas.1012
 call msg_put( lfclsv, 1, size(lfclsv))
*i,setdas.1207
 call msg_get( lfclsv, 1, size(lfclsv))
```



```
*/
*/
*d,setdas.1363,setdas.1370
 if( allocated(ara) ) then deallocate( &
  & ara, den, rkpl, tfc,
& jfq, lfcl, lfclsv,
& ddm, ddn, dec, dxc,
& pan, pcc, pwb, rho,
& isef, laj, lcaj, lse,
                     vol, xlk, yla, &
nbal, npsw, nsl, ntbb, &
dxd, febl, flx, fme, fso, pac, &
shsd, stt, sump, swwfa,wns, wwfa, &
                     maze, ndpf, ndr, nhsd &
  &)
*/
*/
*/
*/
*/
*open,mcrun.F90
* /
*i,mcrun.72
do 50 i=1,mxa
50 lfclsv(i) = lfcl(i)
*/
*/
,
*/
*/
*open,isourc.F90
*i,isourc.32
  nsz = nsa
*/
.
*/
*/
*/
*open,crit1 mod.F90
*/
*/
*d,crit1_.75
  zz(i) = sumk(i)/nsrck
zz(i) = sumk(i)/wt0/nsz
1
  if(idum(5).eq.100)write(jtty,*)'kcalc; calculating 3 keffs; nps,nsz,wt0: ',nps,nsz,wt0
*i,crit1_.222
 nsz = nsa
*/
*i,crit1 .232
 if(idum(5).eq.100)write(jtty,*)'kcalc; checking source overrun; kcy,nps,kcsf: ',kcy,nps,kcsf
*i,crit1_.240
  nsz = nsa
*/
*/
*/
*open,msqcon.F90
*i,msgcon.462
  call msg_get( n )
                 nsz = nsz+n
```



```
*/
*/
*/
*/
*open,msgtsk.F90
* /
*i,msgtsk.193
 nsz = 0
*/
*i,msgtsk.386
  call msg_put( nsz )
*/
*/
*open, hstory.F90
*
*i,hstory.60
if(idum(4).gt.0)then
    do 35 i=1,mxa
35 lfcl(i) = 0
 else
do 36 i=1,mxa
36 lfcl(i) = lfclsv(i)
 endif
*d,hstory.191
   if( lfclsv(icl)/=0 ) then
*/
*i,hstory.477
    if(idum(4).gt.0)then
    do 330 i=1,mxa
    lfcl(i) = 0
330
    else
do 340 i=1,mxa
lfcl(i) = lfclsv(i)
340
     endif
*/
*/
*/
*open,startp.F90
*/
*i,startp.19
idum(4) = idum(3)
if(idum(4).gt.0)then
    do 10 i=1,mxa
10 lfcl(i) = 0
    olse

else
do 20 i=1,mxa
20 lfcl(i) = lfclsv(i)
 endif
*/
*/
*/
*open,acetot.F90
* /
*d,acetot.205
! if( lfcl(icl)==0 ) return
if( lfclsv(icl)==0 ) return
```



```
*/
*/
*open,colidn.F90
*i,colidn.12
  kk = 0
*/
*i,colidn.40
  ! following just to get the collision density estimate of keff at in-superhistory cycles:
  ms = m
  if(idum(4).lt.0)stop 123
  if(nsr.eq.71.and.idum(4).ne.0)then
fm = 0.
ab = 0.
af = 0.
      do m=jmd(mk), jmd(mk+1)-1
        t1 = fme(m)*rtc(krtc+8,lme(1,m))
fm = fm+t1*rtc(krtc+10,lme(1,m))
        af = af+t1
         ab = ab+fme(m)*rtc(krtc+3,lme(1,m))
     end do
      ! get collision estimate of prompt neutron removal lifetime.
     rh = rho(icl)
     a = wgt*rh/(rh*totm+abs(ra_overs()))
     rlttc(1,1) = rlttc(1,1)+(ab+af)*a*tme
rlttc(1,2) = rlttc(1,2)+(ab+af)*a
      ! increment k tallies and summary tables if fission is possible.
     if(fm.ne.0.)then
        xf = fm*a
if( krflg/=2 ) sumktc(1)=sumktc(1)+xf
     endif
  endif
  m = ms
*d,colidn.64
  if( nsr==71.and.idum(4)==0 ) then
*i,colidn.70
  ! following just to get the absorption estimate of keff at in-superhistory cycles:
  if(idum(4).gt.0)then
     [1dum(4).gt.0)then
! analog capture.
if(wcl(1)==0. .or. erg<=emcf(1) ) then
t1 = rtc(krtc+3,iex)+rtc(krtc+8,iex)
r = rang()*rtc(krtc+5,iex)
if(t1<r) go to 150
i = 2</pre>
              3
         i =
         if(r >= rtc(krtc+3, iex)) i = 4
        II( r>=rcc(krc+3,lex) ) 1 = 4
pan(kpan+1,i,mpan) = pan(kpan+1,i,mpan)+wgt
if( nsr==71 ) then
rlttc(2,1) = rlttc(2,1)+wgt*tme
rlttc(2,2) = rlttc(2,2)+wgt
if( i==4 ) rlttc(4,1) = rlttc(4,1)+wgt*tme
sumktc(2) = sumktc(2)+wgt*rtc(krtc+10,iex)*rtc(krtc+8,iex)/t1
endif
         endif
     go to 150
endif
      ! implicit capture
     t8 = wgt/rtc(krtc+5,iex)
t1 = t8*(rtc(krtc+3,iex)+rtc(krtc+8,iex))
     t1 = t8*(rtc(krtc+3,iex)+rtc(krtc+8,iex))
if( nsr==71 ) then
rlttc(2,1) = rlttc(2,1)+t1*tme
rlttc(2,2) = rlttc(2,2)+t1
rlttc(4,1) = rlttc(4,1)+t8*rtc(krtc+8,iex)*tme
sumktc(2) = sumktc(2) +t8*rtc(krtc+10,iex)*rtc(krtc+8,iex)
pan(kpan+1,3,mpan) = pan(kpan+1,3,mpan)+t8*rtc(krtc+3,iex)
pan(kpan+1,4,mpan) = pan(kpan+1,4,mpan)+t8*rtc(krtc+8,iex)
      endif
150 continue
  endif
*/
*d,colidn.74
       t1 = rtc(krtc+3,iex)+rtc(krtc+8,iex)
1
     rt = rtc(krtc+8,iex)
if(idum(4).gt.0)rt = zero
      t1 = rtc(krtc+3,iex)+rt
*d,colidn.83
```



if(nsr==71.and.idum(4)==0) then

```
*d,colidn.94
! t1 = t8*(rtc(krtc+3,iex)+rtc(krtc+8,iex))
rt = rtc(krtc+8,iex)
if(idum(4).gt.0)rt = zero
t1 = t8*(rtc(krtc+3,iex)+rt)
*d,colidn.95
  if( nsr==71.and.idum(4)==0 ) then
*d,colidn.152
  call acecol(0,kk)
*/
*i,colidn.263
      if(kk.eq.1)nsz = nsz + 1
*/
*i,colidn.287
1
1
     stop because we should never be here and there is a call to colidk.
 stop 11
1
*/
*i,colidn.309
  if(kk.eq.1)nsz = nsz + 1
*/
*/
*/
*open,colidk.F90
* /
*d,colidk.57
 n1 = xf/(rkk**(idum(3)+1))+rang()
+ /
*d,colidk.112
        call acecol(2,0)
*/
*/
*/
*/
*open,acecol.F90
'
'
*d,acecol.4
subroutine acecol(jq,kk)
*i,acecol.11
  kk = 0
*/
*d,acecol.28
     sr = rtc(krtc+4,iex)-rtc(krtc+3,iex)-rtc(krtc+8,iex)-el
1
    sr = rtc(krtc+3,iex)-rtc(krtc+3,iex)-rtc(
rt = rtc(krtc+3,iex)
if(idum(4).gt.0)rt = zero
sr = rtc(krtc+4,iex)-rtc(krtc+3,iex)-rt-el
*d,acecol.35
1
       sr = rtc(krtc+4,iex)-rtc(krtc+3,iex)-rtc(krtc+8,iex)-el
      rt = rtc(krtc+8,iex)
if(idum(4).gt.0)rt = zero
sr = rtc(krtc+4,iex)-rtc(krtc+3,iex)-rt-el
*i,acecol.85
    if(rtc(krtc+8,iex).eq.zero)stop 80
*/
*i,acecol.137
  if(jq.eq.0)then
    if(ntyn.ge.18.and.ntyn.le.21)then
    if(mtp.lt.18.or.mtp.gt.21)stop 137
      if(idum(4).gt.0)then
idum(4) = idum(4) - 1
        kk = 1
      else
        stop 19
      endif
    endif
```



```
endif
*/
*/
*open,bankit.F90
*/
*i,bankit.32
 spare(1) = idum(4) + thousandth
*i,bankit.234
 idum(4) = spare(1)
*/
*open,mapmaz.F90
*/
*/
*d,mapmaz.78
 if( lfclsv(ic(lv))/=0 ) then
*/
*/
 end of patch
```