```
!=====================================================================!
!                                                                     !
!   Software Name : HEC-MW Library for PC-cluster                     !
!         Version : 2.5                                               !
!                                                                     !
!     Last Update : 2006/06/01                                        !
!        Category : Linear Solver                                     !
!                                                                     !
!            Written by Kengo Nakajima (Univ. of Tokyo)               !
!                                                                     !
!     Contact address :  IIS,The University of Tokyo RSS21 project   !
!                                                                     !
!     "Structural Analysis System for General-purpose Coupling       !
!      Simulations Using High End Computing Middleware (HEC-MW)"      !
!                                                                     !
!=====================================================================!

      module hecmw_solver_33
      contains
!
!C***
!C*** hecmw_solve_33
!C***
!
        subroutine hecmw_solve_33 (hecMESH, hecMAT)

        use hecmw_util
        use hecmw_solver_CG_33
        use hecmw_solver_BiCGSTAB_33
        use hecmw_solver_GMRES_33
        use hecmw_solver_GPBiCG_33
        use hecmw_solver_BLCG_33
        use hecmw_solver_BLBiCGSTAB_33
        use hecmw_solver_BLGPBiCG_33
        use m_hecmw_solve_error
        use m_hecmw_comm_f
```

```
37          use hecmw_matrix_ass
38          use hecmw_matrix_contact
39
40          implicit none
41
42          type (hecmwST_matrix)     :: hecMAT
43          type (hecmwST_local_mesh) :: hecMESH
44
45          integer(kind=kint) :: ERROR
46          real(kind=kreal), dimension(3,3) :: ALU
47          real(kind=kreal), dimension(3)   :: PW
48
49          integer(kind=kint) :: ITER, METHOD, PRECOND, NSET
50          integer(kind=kint) :: iterPREmax, ii, i, j, k
51
52          real(kind=kreal) :: RESID, SIGMA_DIAG, THRESH, FILTER
53
54          integer(kind=kint) :: ITERlog, TIMElog
55          real(kind=kreal) :: TIME_setup, TIME_comm, TIME_soltot, TR
56          real(kind=kreal),    dimension(1) :: RHS
57          integer (kind=kint), dimension(1) :: IFLAG
58
59          integer(kind=kint) :: NREST,NPLU,iS0,iS,inum,k0
60          real(kind=kreal)   :: SIGMA,S1_time,E1_time,TIME_sol
61
62  !C===
63  !C +------------+
64  !C | PARAMETERs |
65  !C +------------+
66  !C===
67          ITER      = hecMAT%Iarray(1)
68          METHOD    = hecMAT%Iarray(2)
69          PRECOND   = hecMAT%Iarray(3)
70          NSET      = hecMAT%Iarray(4)
71          iterPREmax= hecMAT%Iarray(5)
72          NREST     = hecMAT%Iarray(6)
```

```
 73
 74           ITERlog= hecMAT%Iarray(21)
 75           TIMElog= hecMAT%Iarray(22)
 76
 77           TIME_setup  = 0.d0
 78           TIME_comm   = 0.d0
 79           TIME_soltot = 0.d0
 80
 81           RESID     = hecMAT%Rarray(1)
 82           SIGMA_DIAG= hecMAT%Rarray(2)
 83           SIGMA     = hecMAT%Rarray(3)
 84
 85           THRESH= hecMAT%Rarray(4)
 86           FILTER= hecMAT%Rarray(5)
 87
 88           if (iterPREmax.lt.1) iterPREmax= 1
 89           if (iterPREmax.gt.4) iterPREmax= 4
 90
 91   !C===
 92   !C +-------------+
 93   !C | ERROR CHECK |
 94   !C +-------------+
 95   !C===
 96           ERROR= 0
 97
 98   !C
 99   !C-- ZERO RHS norm
100           RHS(1)= 0.d0
101           do i= 1, hecMAT%N
102             RHS(1)= RHS(1) + hecMAT%B(3*i-2)**2 + hecMAT%B(3*i-1)**2       &
103          &                 + hecMAT%B(3*i  )**2
104           enddo
105           if (hecMESH%mpc%n_mpc > 0) then
106             do i= 1, hecMESH%mpc%n_mpc
107               RHS(1)= RHS(1) + hecMESH%mpc%mpc_const(i)**2
108             enddo
```

```
109           endif
110           call hecmw_allreduce_R (hecMESH, RHS, 1, hecmw_sum)
111
112           if (RHS(1).eq.0.d0) then
113             ERROR= 2002
114             call hecmw_solve_error (hecMESH, ERROR)
115           endif
116
117   !C
118   !C-- ZERO DIAGONAL component
119           IFLAG(1)= 0
120           do i= 1, hecMAT%N
121             if (dabs(hecMAT%D(9*i-8)).eq.0.d0) IFLAG(1)= 1
122             if (dabs(hecMAT%D(9*i-4)).eq.0.d0) IFLAG(1)= 1
123             if (dabs(hecMAT%D(9*i  )).eq.0.d0) IFLAG(1)= 1
124           enddo
125
126           call hecmw_allreduce_I (hecMESH, IFLAG, 1, hecmw_sum)
127           if (IFLAG(1).ne.0) then
128             ERROR= 2001
129             call hecmw_solve_error (hecMESH, ERROR)
130           endif
131
132   !C
133   !C-- INCONSISTENT SOLVER/PRECONDITIONING
134           IFLAG(1)= 0
135           if (METHOD.le.0 .or. METHOD.ge.5)    IFLAG(1)= 1
136           if (PRECOND.le.0 .or. PRECOND.gt.21) IFLAG(1)= 1
137
138           call hecmw_allreduce_I (hecMESH, IFLAG, 1, hecmw_sum)
139           if (IFLAG(1).ne.0) then
140             ERROR= 1001
141             call hecmw_solve_error (hecMESH, ERROR)
142           endif
143
144           IFLAG(1)= 1
```

```
145           if (METHOD.eq.1) then
146             if (PRECOND.eq. 1) IFLAG(1)= 0
147             if (PRECOND.eq. 2) IFLAG(1)= 0
148             if (PRECOND.eq. 3) IFLAG(1)= 0
149             if (PRECOND.eq.10) IFLAG(1)= 0
150             if (PRECOND.eq.11) IFLAG(1)= 0
151             if (PRECOND.eq.12) IFLAG(1)= 0
152           endif
153
154           if (METHOD.eq.2) then
155             if (PRECOND.eq. 1) IFLAG(1)= 0
156             if (PRECOND.eq. 2) IFLAG(1)= 0
157             if (PRECOND.eq. 3) IFLAG(1)= 0
158             if (PRECOND.eq.10) IFLAG(1)= 0
159             if (PRECOND.eq.11) IFLAG(1)= 0
160             if (PRECOND.eq.12) IFLAG(1)= 0
161           endif
162
163           if (METHOD.eq.3) then
164             if (PRECOND.eq. 1) IFLAG(1)= 0
165             if (PRECOND.eq. 2) IFLAG(1)= 0
166             if (PRECOND.eq. 3) IFLAG(1)= 0
167           endif
168
169           if (METHOD.eq.4) then
170             if (PRECOND.eq. 1) IFLAG(1)= 0
171             if (PRECOND.eq. 2) IFLAG(1)= 0
172             if (PRECOND.eq. 3) IFLAG(1)= 0
173             if (PRECOND.eq.10) IFLAG(1)= 0
174             if (PRECOND.eq.11) IFLAG(1)= 0
175             if (PRECOND.eq.12) IFLAG(1)= 0
176           endif
177
178           if (IFLAG(1).ne.0) then
179             ERROR= 1001
180             call hecmw_solve_error (hecMESH, ERROR)
```

```
181           endif
182
183   !C===
184   !C +-----------+
185   !C | BLOCK LUs |
186   !C +-----------+
187   !C===
188           if (.not.associated(hecMAT%ALU).and.PRECOND.lt.10) NSET=  0
189           if (     associated(hecMAT%ALU).and.PRECOND.lt.10) NSET= -1
190
191           S1_time= HECMW_WTIME()
192           if (NSET.eq.0 .and. PRECOND.lt.10) then
193             allocate (hecMAT%ALU(9*hecMAT%N))
194           endif
195           if (NSET.le.0 .and. PRECOND.lt.10) then
196             hecMAT%ALU  = 0.d0
197
198             do ii= 1, hecMAT%N
199               hecMAT%ALU(9*ii-8) = hecMAT%D(9*ii-8)
200               hecMAT%ALU(9*ii-7) = hecMAT%D(9*ii-7)
201               hecMAT%ALU(9*ii-6) = hecMAT%D(9*ii-6)
202               hecMAT%ALU(9*ii-5) = hecMAT%D(9*ii-5)
203               hecMAT%ALU(9*ii-4) = hecMAT%D(9*ii-4)
204               hecMAT%ALU(9*ii-3) = hecMAT%D(9*ii-3)
205               hecMAT%ALU(9*ii-2) = hecMAT%D(9*ii-2)
206               hecMAT%ALU(9*ii-1) = hecMAT%D(9*ii-1)
207               hecMAT%ALU(9*ii  ) = hecMAT%D(9*ii  )
208             enddo
209
210             if (hecMAT%cmat%n_val.gt.0) then
211               do k= 1, hecMAT%cmat%n_val
212                 if (hecMAT%cmat%pair(k)%i.ne.hecMAT%cmat%pair(k)%j) cycle
213                 ii = hecMAT%cmat%pair(k)%i
214                 hecMAT%ALU(9*ii-8) = hecMAT%ALU(9*ii-8) + hecMAT%cmat%pair(k)%val(1, 1)
215                 hecMAT%ALU(9*ii-7) = hecMAT%ALU(9*ii-7) + hecMAT%cmat%pair(k)%val(1, 2)
216                  hecMAT%ALU(9*ii-6) = hecMAT%ALU(9*ii-6) + hecMAT%cmat%pair(k)%val(1,
```

```
217    3)
218                  hecMAT%ALU(9*ii-5) = hecMAT%ALU(9*ii-5) + hecMAT%cmat%pair(k)%val(2,
219    1)
220                  hecMAT%ALU(9*ii-4) = hecMAT%ALU(9*ii-4) + hecMAT%cmat%pair(k)%val(2,
221    2)
222                  hecMAT%ALU(9*ii-3) = hecMAT%ALU(9*ii-3) + hecMAT%cmat%pair(k)%val(2,
223    3)
224                  hecMAT%ALU(9*ii-2) = hecMAT%ALU(9*ii-2) + hecMAT%cmat%pair(k)%val(3,
225    1)
226                  hecMAT%ALU(9*ii-1) = hecMAT%ALU(9*ii-1) + hecMAT%cmat%pair(k)%val(3,
227    2)
228                  hecMAT%ALU(9*ii  ) = hecMAT%ALU(9*ii  ) + hecMAT%cmat%pair(k)%val(3,
229    3)
230                enddo
231
232                call hecmw_cmat_LU( hecMAT )
233
234              endif
235
236              do ii= 1, hecMAT%N
237                ALU(1,1)= hecMAT%ALU(9*ii-8)
238                ALU(1,2)= hecMAT%ALU(9*ii-7)
239                ALU(1,3)= hecMAT%ALU(9*ii-6)
240                ALU(2,1)= hecMAT%ALU(9*ii-5)
241                ALU(2,2)= hecMAT%ALU(9*ii-4)
242                ALU(2,3)= hecMAT%ALU(9*ii-3)
243                ALU(3,1)= hecMAT%ALU(9*ii-2)
244                ALU(3,2)= hecMAT%ALU(9*ii-1)
245                ALU(3,3)= hecMAT%ALU(9*ii  )
246                do k= 1, 3
247                  ALU(k,k)= 1.d0/ALU(k,k)
248                  do i= k+1, 3
249                    ALU(i,k)= ALU(i,k) * ALU(k,k)
250                    do j= k+1, 3
251                      PW(j)= ALU(i,j) - ALU(i,k)*ALU(k,j)
252                    enddo
```

```
253                   do j= k+1, 3
254                     ALU(i,j)= PW(j)
255                   enddo
256                 enddo
257               enddo
258               hecMAT%ALU(9*ii-8)= ALU(1,1)
259               hecMAT%ALU(9*ii-7)= ALU(1,2)
260               hecMAT%ALU(9*ii-6)= ALU(1,3)
261               hecMAT%ALU(9*ii-5)= ALU(2,1)
262               hecMAT%ALU(9*ii-4)= ALU(2,2)
263               hecMAT%ALU(9*ii-3)= ALU(2,3)
264               hecMAT%ALU(9*ii-2)= ALU(3,1)
265               hecMAT%ALU(9*ii-1)= ALU(3,2)
266               hecMAT%ALU(9*ii  )= ALU(3,3)
267             enddo
268           endif
269
270           E1_time= HECMW_WTIME()
271           TIME_setup= TIME_setup + E1_time - S1_time
272
273   !C===
274   !C +------------------+
275   !C | ITERATIVE solver |
276   !C +------------------+
277   !C===
278
279   !C
280   !C-- CG
281         if (METHOD.eq.1 .and. PRECOND.lt.10) then
282           if (hecMESH%my_rank.eq.0 .and. (ITERlog.eq.1 .or. TIMElog.eq.1)) then
283             if (PRECOND.eq.1) write (*,'(a,i3)') '### 3x3 B-IC-CG  (0)',  &
284         &                                         iterPREmax
285             if (PRECOND.eq.2) write (*,'(a,i3)') '### 3x3 B-SSOR-CG(0)',  &
286         &                                         iterPREmax
287             if (PRECOND.eq.3) write (*,'(a,i3)') '### 3x3 B-scale-CG  ',  &
288         &                                         iterPREmax
```

```
289           endif
290
291           call hecmw_solve_CG_33( hecMESH,  hecMAT, ITER, RESID, ERROR,   &
292         &                         TIME_setup, TIME_sol, TIME_comm )
293         endif
294
295   !C
296   !C-- BiCGSTAB
297         if (METHOD.eq.2 .and. PRECOND.lt.10) then
298           if (hecMESH%my_rank.eq.0 .and. (ITERlog.eq.1 .or. TIMElog.eq.1)) then
299             if (PRECOND.eq.1) write (*,'(a,i3)')                          &
300         &                   '### 3x3 B-ILU-BiCGSTAB (0)', iterPREmax
301             if (PRECOND.eq.2) write (*,'(a,i3)')                          &
302         &                   '### 3x3 B-SSOR-BiCGSTAB(0)', iterPREmax
303             if (PRECOND.eq.3) write (*,'(a,i3)')                          &
304         &                   '### 3x3 B-scale-BiCGSTAB  ', iterPREmax
305           endif
306
307           call hecmw_solve_BiCGSTAB_33( hecMESH,  hecMAT, ITER, RESID, ERROR, &
308         &                               TIME_setup, TIME_sol, TIME_comm )
309         endif
310
311   !C
312   !C-- GMRES
313         if (METHOD.eq.3 .and. PRECOND.lt.10) then
314           ! imposing MPC by penalty
315           call hecmw_mat_ass_equation ( hecMESH, hecMAT )
316
317           if (hecMESH%my_rank.eq.0 .and. (ITERlog.eq.1 .or. TIMElog.eq.1)) then
318             if (PRECOND.eq.1) write (*,'(a,i3)')                          &
319         &                   '### 3x3 B-ILU-GMRES (0)', iterPREmax
320             if (PRECOND.eq.2) write (*,'(a,i3)')                          &
321         &                   '### 3x3 B-SSOR-GMRES(0)', iterPREmax
322             if (PRECOND.eq.3) write (*,'(a,i3)')                          &
323         &                   '### 3x3 B-scale-GMRES  ', iterPREmax
324           endif
```

```
325
326           call hecmw_solve_GMRES_33                                    &
327         &     ( hecMAT%N, hecMAT%NP, hecMAT%NPL, hecMAT%NPU,           &
328         &       hecMAT%D, hecMAT%AL, hecMAT%indexL, hecMAT%itemL,      &
329         &                 hecMAT%AU, hecMAT%indexU, hecMAT%itemU,      &
330         &       hecMAT%B, hecMAT%X,  hecMAT%ALU, RESID, ITER,          &
331         &       ERROR,    hecMESH%my_rank,                             &
332         &       hecMESH%n_neighbor_pe, hecMESH%neighbor_pe,            &
333         &       hecMESH%import_index, hecMESH%import_item,             &
334         &       hecMESH%export_index, hecMESH%export_item,             &
335         &       hecMESH%MPI_COMM, PRECOND, iterPREmax, NREST,          &
336         &       TIME_setup, TIME_sol, TIME_comm, ITERlog)
337          endif
338
339   !C
340   !C-- GPBiCG
341          if (METHOD.eq.4 .and. PRECOND.lt.10) then
342            ! imposing MPC by penalty
343            call hecmw_mat_ass_equation ( hecMESH, hecMAT )
344
345            if (hecMESH%my_rank.eq.0 .and. (ITERlog.eq.1 .or. TIMElog.eq.1)) then
346              if (PRECOND.eq.1) write (*,'(a,i3)')                       &
347         &                    '### 3x3 B-ILU-GPBiCG   (0)', iterPREmax
348              if (PRECOND.eq.2) write (*,'(a,i3)')                       &
349         &                    '### 3x3 B-SSOR-GPBiCG  (0)', iterPREmax
350              if (PRECOND.eq.3) write (*,'(a,i3)')                       &
351         &                    '### 3x3 B-scale-GPBiCG    ', iterPREmax
352            endif
353
354            call hecmw_solve_GPBiCG_33                                 &
355         &     ( hecMAT%N, hecMAT%NP, hecMAT%NPL, hecMAT%NPU,           &
356         &       hecMAT%D, hecMAT%AL, hecMAT%indexL, hecMAT%itemL,      &
357         &                 hecMAT%AU, hecMAT%indexU, hecMAT%itemU,      &
358         &       hecMAT%B, hecMAT%X,  hecMAT%ALU, RESID, ITER,          &
359         &       ERROR,    hecMESH%my_rank,                             &
360         &       hecMESH%n_neighbor_pe, hecMESH%neighbor_pe,            &
```

```
361         &       hecMESH%import_index, hecMESH%import_item,                  &
362         &       hecMESH%export_index, hecMESH%export_item,                  &
363         &       hecMESH%MPI_COMM, PRECOND, iterPREmax,                      &
364         &       TIME_setup, TIME_sol, TIME_comm, ITERlog)
365          endif
366
367     !C
368     !C-- CG-1/2
369          if (METHOD.eq.1 .and. (PRECOND.ge.10.and.PRECOND.le.20)) then
370            ! imposing MPC by penalty
371            call hecmw_mat_ass_equation ( hecMESH, hecMAT )
372
373            if (hecMESH%my_rank.eq.0 .and. (ITERlog.eq.1 .or. TIMElog.eq.1)) then
374              if (PRECOND.eq.10) write (*,'(a)') '### 3x3 B-IC-CG  (0)'
375              if (PRECOND.eq.11) write (*,'(a)') '### 3x3 B-IC-CG  (1)'
376              if (PRECOND.eq.12) write (*,'(a)') '### 3x3 B-IC-CG  (2)'
377            endif
378
379            SIGMA     = 1.d0
380            SIGMA_DIAG= 1.d0
381            call hecmw_solve_BLCG_33                                        &
382         &     ( hecMAT%N, hecMAT%NP, hecMAT%NPL, hecMAT%NPU,              &
383         &       hecMAT%D, hecMAT%AL, hecMAT%indexL, hecMAT%itemL,         &
384         &                 hecMAT%AU, hecMAT%indexU, hecMAT%itemU,         &
385         &       hecMAT%B, hecMAT%X,  RESID, SIGMA, SIGMA_DIAG,            &
386         &       ITER, ERROR,  hecMESH%my_rank,                            &
387         &       hecMESH%n_neighbor_pe, hecMESH%neighbor_pe,               &
388         &       hecMESH%import_index, hecMESH%import_item,                &
389         &       hecMESH%export_index, hecMESH%export_item,                &
390         &       hecMESH%MPI_COMM, PRECOND, iterPREmax,                    &
391         &       TIME_setup, TIME_sol, TIME_comm, ITERlog)
392          endif
393
394     !C
395     !C-- BiCGSTAB-1/2
396          if (METHOD.eq.2 .and. (PRECOND.ge.10.and.PRECOND.le.20)) then
```

```
397           ! imposing MPC by penalty
398           call hecmw_mat_ass_equation ( hecMESH, hecMAT )
399
400           if (hecMESH%my_rank.eq.0 .and. (ITERlog.eq.1 .or. TIMElog.eq.1)) then
401             if (PRECOND.eq.10) write (*,'(a)')                              &
402         &                      '### 3x3 B-IlU-BiCGSTAB (0)'
403             if (PRECOND.eq.11) write (*,'(a)')                              &
404         &                      '### 3x3 B-IlU-BiCGSTAB (1)'
405             if (PRECOND.eq.12) write (*,'(a)')                              &
406         &                      '### 3x3 B-IlU-BiCGSTAB (2)'
407           endif
408
409           SIGMA     = 1.d0
410           SIGMA_DIAG= 1.d0
411           call hecmw_solve_BLBiCGSTAB_33                                    &
412         &     ( hecMAT%N, hecMAT%NP, hecMAT%NPL, hecMAT%NPU,                &
413         &       hecMAT%D, hecMAT%AL, hecMAT%indexL, hecMAT%itemL,           &
414         &                 hecMAT%AU, hecMAT%indexU, hecMAT%itemU,           &
415         &       hecMAT%B, hecMAT%X,  RESID, SIGMA, SIGMA_DIAG,              &
416         &       ITER, ERROR,  hecMESH%my_rank,                              &
417         &       hecMESH%n_neighbor_pe, hecMESH%neighbor_pe,                 &
418         &       hecMESH%import_index, hecMESH%import_item,                  &
419         &       hecMESH%export_index, hecMESH%export_item,                  &
420         &       hecMESH%MPI_COMM, PRECOND, iterPREmax,                      &
421         &       TIME_setup, TIME_sol, TIME_comm, ITERlog)
422         endif
423
424   !C
425   !C-- GPBiCG-1/2
426         if (METHOD.eq.4 .and. (PRECOND.ge.10.and.PRECOND.le.20)) then
427           ! imposing MPC by penalty
428           call hecmw_mat_ass_equation ( hecMESH, hecMAT )
429
430           if (hecMESH%my_rank.eq.0 .and. (ITERlog.eq.1 .or. TIMElog.eq.1)) then
431             if (PRECOND.eq.10) write (*,'(a)') '### 3x3 B-ILU-GPBiCG (0)'
432             if (PRECOND.eq.11) write (*,'(a)') '### 3x3 B-ILU-GPBiCG (1)'
```

```
433             if (PRECOND.eq.12) write (*,'(a)') '### 3x3 B-ILU-GPBiCG (2)'
434           endif
435
436           SIGMA     = 1.d0
437           SIGMA_DIAG= 1.d0
438           call hecmw_solve_BLGPBiCG_33                                    &
439         &     ( hecMAT%N, hecMAT%NP, hecMAT%NPL, hecMAT%NPU,              &
440         &       hecMAT%D, hecMAT%AL, hecMAT%indexL, hecMAT%itemL,         &
441         &                 hecMAT%AU, hecMAT%indexU, hecMAT%itemU,         &
442         &       hecMAT%B, hecMAT%X,  RESID, SIGMA, SIGMA_DIAG,            &
443         &       ITER, ERROR,  hecMESH%my_rank,                            &
444         &       hecMESH%n_neighbor_pe, hecMESH%neighbor_pe,               &
445         &       hecMESH%import_index, hecMESH%import_item,                &
446         &       hecMESH%export_index, hecMESH%export_item,                &
447         &       hecMESH%MPI_COMM, PRECOND, iterPREmax,                    &
448         &       TIME_setup, TIME_sol, TIME_comm, ITERlog)
449         endif
450
451         if (RESID.gt.hecMAT%Rarray(1)) then
452           call hecmw_solve_error (hecMESH, 3001)
453         endif
454
455         if (hecMESH%my_rank.eq.0 .and. TIMElog.eq.1) then
456           TR= (TIME_sol-TIME_comm)/(TIME_sol+1.d-24)*100.d0
457           write (*,'(/a)')          '### summary of linear solver'
458           write (*,'(i10,a, 1pe16.6)')      ITER, ' iterations  ', RESID
459           write (*,'(a, 1pe16.6 )') '    set-up time      : ', TIME_setup
460           write (*,'(a, 1pe16.6 )') '    solver time      : ', TIME_sol
461           write (*,'(a, 1pe16.6 )') '    solver/comm time : ', TIME_comm
462           write (*,'(a, 1pe16.6/)') '    work ratio (%)   : ', TR
463         endif
464
465         if (hecMAT%cmat%n_val.gt.0) then
466           call hecmw_cmat_LU_free( hecMAT )
467         endif
468
```

```
469         end subroutine hecmw_solve_33
470         end module hecmw_solver_33
```