Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

E.D. de Goede, J.H.M. ten Thije Boonkkamp

NUMVEC FORTRAN library manual
Chapter: Partial differential equations
Routine: OEHPC

Department of Numerical Mathematics Report NM-R8915 August

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

NUMVEC FORTRAN Library manual

Chapter: Partial Differential Equations

Routine: OEHPC

E.D. de Goede and J.H.M. ten Thije Boonkkamp
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

This document describes the NUMVEC FORTRAN Library routine OEHPC, which solves the two-dimensional, time-dependent, incompressible Navier-Stokes equations on a rectangular domain.

1980 Mathematics subject classification (1985 revision): 65V05, 65M05, 76DXX.
1982 CR Categories: 5.17.

Keywords & Phrases: software, Navier-Stokes equations, odd-even hopscotch method, pressure correction method.

Note: The implementation is available in auto-vectorizable ANSI FORTRAN 77.

Report NM-R8915
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

OEHPC - NUMVEC FORTRAN Library Routine Document

1. Purpose

OEHPC computes the solution of the two-dimensional, time-dependent, incompressible Navier-Stokes equations on a rectangular domain. For the time-integration the odd-even hopscotch method is combined with an efficient Poisson-solver. The resulting method is called the odd-even hopscotch pressure correction (OEHPC) method.

2. Specification

```
      SUBROUTINE  OEHPC (NXF, NYF, NXC, NYC, LEVELS, NM, STARTL,
     +                   LENX, LENY, NT, TAU, REY, NCYCLE, NY1, NY2, TOL,
     +                   U, V, P, IFAIL,
     +                   WORK1, WORK2, WORK3,
     +                   UN, UE, US, UW, VN, VE, VS, VW)
C      INTEGER    NXF, NYF, NXC, NYC, LEVELS, NM, STARTL, NT, NCYCLE, NY1, NY2,
C     +           WORK3(NXF*NYF+1: NM+4*LEVELS), IFAIL
C      REAL       LENX, LENY, TAU, REY, TOL,
C     +           U(1:NXF,1:NYF), V(1:NXF,0:NYF), P(1:NXF,1:NYF),
C     +           WORK1(1:2*NM), WORK2((-NXF+1)/2:NXF*NYF/2+3*NXF+6*NYF),
C     +           UN, UE, US, UW, VN, VE, VS, VW
C      EXTERNAL   UN, UE, US, UW, VN, VE, VS, VW
```

3. Description

OEHPC computes on a rectangular domain the solution of the two-dimensional, time-dependent, incompressible Navier-Stokes equations

$$u_t = -u u_x - v u_y + (u_{xx} + u_{yy}) / \mathrm{Re} - p_x$$
$$v_t = -u v_x - v v_y + (v_{xx} + v_{yy}) / \mathrm{Re} - p_y ,$$
$$u_x + v_y = 0$$

with Re denoting the Reynolds number, p the pressure and u and v the velocity components in x- and y-direction, respectively. The time-integration technique used is the odd-even hopscotch (OEH) method. In order to decouple the pressure computation from the velocity computation, this method is combined with a pressure correction approach. The resulting method is called the odd-even hopscotch pressure correction (OEHPC) method, and requires for the pressure computation the solution of the Poisson-equation at each time step. A detailed discussion of the OEHPC method is presented in [6].

For the space discretization standard second-order central differences are used on a uniform, staggered grid. Let us now assume that the grid has a chess-board ordering, i.e., the grid is divided in a set of odd cells (corresponding to the white cells) and a set of even cells (corresponding to the black cells). Then the OEH scheme can be considered as a two-stage splitting method, in which the explicit and implicit Euler method are alternatingly used at the odd and even cells, respectively. For a detailed description of the OEH method, the reader is referred to [3]. The method can be implemented efficiently on vector and parallel computers and has fairly good stability properties. A von Neumann stability analysis of the OEH method, when applied to a linear convection-diffusion equation, shows that the scheme is conditionally stable (see e.g. [7]). When using the OEH method, one should realize that the method is only conditionally consistent for

convection-diffusion problems [7]. However, for many practical (fluid flow) computations, this inconsistency is only of minor importance.

The OEHPC scheme is implemented in such a way that the values in respectively the odd- and even cells are stored in consecutive memory locations (stride 1), which is an advantage for use on vector computers. No data reorderings have to be performed. A fully vectorized version of the OEH method, when applied to the two-dimensional Burgers' equations, is given in [2].

A full multigrid (FMG) method is used for the solution of the Poisson equation. This method is a V-cyclic method, with red-black Gauss-Seidel iteration as relaxation technique, bi-linear interpolation as prolongation operator and half weighting as restriction operator [1]. The Poisson-solver is highly vectorized. An extensive description of FMG methods can be found in e.g. [5].

4. References

[1] D. Barkai and A. Brandt, "Vectorized multigrid poisson solver for the CDC Cyber 205," *Appl. Math. Comp.* **13**, pp. 215-227 (1983).

[2] E.D. de Goede and J.H.M. ten Thije Boonkkamp, "Vectorization of the odd-even hopscotch scheme and the alternating direction implicit scheme for the two-dimensional Burgers' equations," *SIAM J. Sci. Stat. Comput.*, (to appear).

[3] A.R. Gourlay, "Hopscotch: a Fast Second-order Partial Differential Equation Solver," *J. Inst. Math. Applics.* **6**, pp. 375-390 (1970).

[4] Numerical Algorithms Group, *NAG FORTRAN Library manual - mark 11.* 1984.

[5] K. Stüben and U. Trottenberg, "Multigrid methods: fundamental algorithms, model problem analysis and applications," in *Lecture Notes in Mathematics 960*, ed. W. Hackbusch and U. Trottenberg, Springer-Verlag, Berlin (1981).

[6] J.H.M. ten Thije Boonkkamp, "The odd-even hopscotch pressure correction scheme for the incompressible Navier-Stokes equations," *SIAM J. Sci. Stat. Comput.* **9**, pp. 252-270 (1988).

[7] J.H.M. ten Thije Boonkkamp and J.G. Verwer, "On the odd-even hopscotch scheme for the numerical integration of time-dependent partial differential equations," *Appl. Num. Math.* **3**, pp. 183-193 (1987).

5. Parameters

NXF, NYF - INTEGER.
On entry, NXF and NYF must specify the number of cells in the horizontal and vertical direction, respectively, on the finest grid in the full multigrid method.
Unchanged on exit.

NXC, NYC - INTEGER.
On entry, NXC and NYC must specify the number of cells in the horizontal and vertical direction, respectively, on the coarsest grid in the full multigrid method.
Unchanged on exit.

LEVELS - INTEGER.
On entry, LEVELS must specify the number of levels in the full multigrid method.
Unchanged on exit.

NM - INTEGER.
On entry, NM must specify an upper bound for the number of grid points on all grids together, in the full multigrid method.
Unchanged on exit.

STARTL - INTEGER.
On entry, STARTL must specify the start level in the full multigrid method. If STARTL = LEVELS, then the full multigrid method starts on the finest grid, and if STARTL = 1 it starts on the coarsest grid.
Unchanged on exit.

Note that the following relations should hold:

$1 \leq$ LEVELS

NXC ≥ 5, NYC ≥ 4, NXC odd and NYC even

NXF ≥ 5, NYF ≥ 4, NXF odd and NYF even

NXF $\times$ NYF $\leq 2 \times (2^{16-1})$ (on the CDC CYBER 205 only)

NXF $= 1+(\text{NXC}-1) \times 2^{\text{LEVELS}-1}$

NYF $=$ NYC $\times 2^{\text{LEVELS}-1}$

NM $\geq 10 \times$ NXF $\times$ NYF$/7$

$1 \leq$ STARTL $\leq$ LEVELS

The program does not check whether these relations are satisfied by the actual values of the parameters.

LENX, LENY - REAL.
On entry, LENX and LENY must specify the length of the computational domain in horizontal and vertical direction, respectively.
Unchanged on exit.

NT - INTEGER.
On entry, NT must specify the number of time steps.
Unchanged on exit.

TAU - REAL.
On entry, TAU must specify the time step used in the OEH method. Due to the conditional stability of the OEH method TAU has to satisfy the following condition:

$$\text{TAU} \leq \min\left\{\frac{\text{LENX}}{\text{NXF}-1}, \frac{\text{LENY}}{\text{NYF}}\right\}.$$

Unchanged on exit.

REY - REAL.
On entry, REY must specify the Reynolds number.
Unchanged on exit.

NCYCLE - INTEGER.
On entry, NCYCLE must specify the maximum number of allowed multigrid iterations. If during the multigrid process either NCYCLE iterations have been performed or the tolerance has been

reached, the multigrid process is stopped.
Unchanged on exit.

NY1, NY2 - INTEGER.
On entry, NY1 and NY2 must specify the number of relaxations before and after the coarse-grid correction in the full multigrid method, respectively.
Unchanged on exit.

TOL - REAL.
On entry, TOL must specify the tolerance desired by the user, where TOL is a bound of the l_2-norm of the residual of the solution for the Poisson equation.
Unchanged on exit.

IFAIL - INTEGER.
Before entry, IFAIL must be assigned a value. For users not familiar with this parameter (described in Chapter P01 of [4]) the recommended value is 0.
Unless the routine detects an error (see Section 6), IFAIL contains 0 on exit.

U - REAL array of DIMENSION (1:NXF,1:NYF).
U(i,j) contains an approximation of the horizontal velocity component at the point $((i-1) \times h, (j-\frac{1}{2}) \times k)$, where h and k denote the grid size in the horizontal and vertical direction, respectively ($h =$ LENX/(NXF-1), $k =$ LENY/NYF).
On entry, the user must specify U at time level $t = 0$.
On exit, U contains the numerical solution at time level $t =$ NT $\times$ TAU.

V - REAL array of DIMENSION (1:NXF,0:NYF).
V(i,j) contains an approximation of the vertical velocity component at the point $((i-\frac{3}{2}) \times h, jk)$.
On entry, the user must specify V at time level $t = 0$.
On exit, V contains the numerical solution at time level $t =$ NT $\times$ TAU.

P - REAL array of DIMENSION (1:NXF,1:NYF).
P(i,j) contains an approximation of the pressure at the point $((i-\frac{3}{2})h, (j-\frac{1}{2})k)$.
On entry, the user must specify P at time level $t = 0$.
On exit, P contains the numerical solution at time level $t =$ NT $\times$ TAU.

WORK1 - REAL array of DIMENSION (2*NM).
WORK2 - REAL array of DIMENSION ((−NXF+1) / 2: NXF*NYF / 2+3*NXF+6*NYF).
WORK3 - INTEGER array of DIMENSION (NXF*NYF+1: NM+4*LEVELS).
Used as work space.

UN, UE, US, UW, VN, VE, VS, VW - REAL FUNCTIONS supplied by the user.
These eight functions define the boundary values of the velocities u and v as follows:

$$\begin{aligned}
\text{UN}(x,t) &= u(x, \text{LENY}, t)\\
\text{UE}(y,t) &= u(\text{LENX}, y, t)\\
\text{US}(x,t) &= u(x, 0, t)\\
\text{UW}(x,t) &= u(0, y, t)\\
\text{VN}(x,t) &= v(x, \text{LENY}, t)\\
\text{VE}(y,t) &= v(\text{LENX}, y, t)\\
\text{VS}(x,t) &= v(x, 0, t)\\
\text{VW}(y,t) &= v(0, y, t)
\end{aligned}$$

These eight functions should be declared EXTERNAL.

6. Error indicators and warnings

Errors detected by the routine:-

IFAIL = 1
The full multigrid process failed to converge within NCYCLE iterations at a particular time step.

7. Auxiliary routines

This routine calls the following NUMVEC FORTRAN Library routines:
MG, INIT1, INIT2, TRANS, RELAX, RESTR, INT2A, INTERP, DEFMX, EXPL, IMPL, BOUNDC and P01AAF.

8. Timing

The following table presents the execution times for the test problem described in Section 13, obtained for the CDC CYBER 205 (SARA, Amsterdam, The Netherlands), the CRAY X-MP/28 (Cray Research, Bracknell, U.K.) and the ALLIANT FX/4 (CWI, Amsterdam, The Netherlands).

mesh size ($h = k$)	Execution times (in seconds)		
	CYBER 205	CRAY X-MP/28	ALLIANT FX/4
1/8	0.2	0.02	0.4
1/16	0.4	0.06	1.3
1/32	1.5	0.23	4.8
1/64	4.9	1.05	19.8
1/128	17.4	5.5	124.3
1/256	92.2	40.0	961.2

9. Storage

There are no internally declared arrays.

10. Accuracy

If the process converges and NCYCLE is large enough, the l_2-norm of the residual of the solution for the Poisson equation becomes less than TOL.

11. Further comments

None.

11.1. Vectorization information

This program is entirely written in ANSI FORTRAN 77 and is optimized for the CDC CYBER 205, the CRAY and the ALLIANT. In order to obtain a good performance, the data-structure has been adapted internally to avoid stride problems (see Section 3). This is done during the initiation phase; the numerical solution is restored according to the normal ordering on exit. Compiler-directives have been used for optimization of the program. Therefore, the performance on the CDC CYBER 205 will increase when e.g., the VAST precompiler (Pacific Sierra Research Corporation) is used.

12. Keywords

Incompressible Navier-Stokes equations.
Odd-even hopscotch method.
Pressure correction method.
Full multigrid method.

13. Example

We consider a steady free convection problem of a fluid in a two-dimensional square cavity. The flow in the cavity is caused by a velocity gradient. The computational domain is [0,1] × [0,1] (thus, LENX = LENY = 1). The boundary conditions are:

u(x, 1, t) = 1.,
u(1, y, t) = 0.,
u(x, 0, t) = 0.,
u(0, y, t) = 0.,
v(x, 1, t) = 0.,
v(1, y, t) = 0.,
v(x, 0, t) = 0.,
v(0, y, t) = 0.

The time integration interval is [0,1] and REY = 1000. The experiment was performed with grid sizes $h = k = 1/8, 1/16, 1/32, 1/64, 1/128, 1/256$ and with the time step TAU = h. The tolerance parameter was set to TOL = 10^{-4}.
For timings see Section 8.

13.1. Program text

```
C
C         OEHPC EXAMPLE PROGRAM TEXT
C         NUMVEC 1989
C
      PROGRAM EXAMPLE
C
      INTEGER NF,MF,NC,MC,LEVELS,NM,STARTL
C
      PARAMETER ( NC = 5 , MC = 4 , LEVELS = 6)
      PARAMETER ( NF = (NC-1)*2**(LEVELS-1) + 1 )
      PARAMETER ( MF = MC*2**(LEVELS-1) )
      PARAMETER ( NM = ( 10 * NF * MF) / 7 )
C
      REAL U(1:NF,1:MF),V(1:NF,0:MF),P(1:NF,1:MF)
      REAL H,K,LENX,LENY,TAU,REY,TOL
      REAL UW,UE,VW,VE,US,UN,VS,VN
      INTEGER NT,NCYCLE,NY1,NY2,IFAIL

C         WORK SPACE:
      REAL WORK1(1:2*NM),WORK2((-NF+1)/2:NF*MF/2+3*NF+6*MF)
      INTEGER WORK3(1:(3*NF*MF)/7+4*LEVELS)
C
      EXTERNAL UW,UE,VW,VE,US,UN,VS,VN
```

```
C
      OPEN ( UNIT = 3 , FILE = 'INPUT' )
      OPEN ( UNIT = 4 , FILE = 'OUTPUT' )

      LENX = 1.
      LENY = 1.
      H = LENX/(NF-1)
      K = LENY/MF
C
      WRITE (4,'('' INTERVAL X-DIRECTION [0, : '',F8.4)')   LENX
      WRITE (4,'('' INTERVAL Y-DIRECTION [0, : '',F8.4)')   LENY
      WRITE (4,'('' MESH-WIDTH IN X-DIRECTION : '',F8.4)')H
      WRITE (4,'('' MESH-WIDTH IN Y-DIRECTION : '',F8.4)')K
C
      READ (3,*) NT
      READ (3,*) TAU
      READ (3,*) REY
      READ (3,*) NCYCLE
      READ (3,*) NY1
      READ (3,*) NY2
      READ (3,*) STARTL
      READ (3,*) TOL

      WRITE (4,'('' REYNOLDS-NUMBER          : '',F8.1)') REY
      WRITE (4,'('' TIME-STEP                : '',F8.4)') TAU
      WRITE (4,'('' NUMBER OF TIME-STEPS     : '',I8)') NT
      WRITE (4,'('' NUMBER OF MG-CYCLES      : '',I8)') NCYCLE
      WRITE (4,'('' START LEVEL              : '',I8)') STARTL
      WRITE (4,'('' TOL                      :  '',E6.1)') TOL
      WRITE (4,'('' ITERATIONS BEFORE CGC    : '',I8)') NY1
      WRITE (4,'('' ITERATIONS AFTER CGC     : '',I8)') NY2
C
C     COMPUTATION OF THE INITIAL VALUES:
      DO 100 J=1,MF-1
       DO 100 I=1,NF
        U(I,J) = 0.
  100 CONTINUE
      DO 110 I=1,NF
       U(I,MF) = 1.
  110 CONTINUE
C
      DO 200 J=0,MF
       DO 200 I=1,NF
        V(I,J) = 0.
  200 CONTINUE
C
      DO 300 J=1,MF
       DO 300 I=1,NF
        P(I,J) = 0.
```

```
  300  CONTINUE
C
       T0 = TIMING(T0)

       CALL OEHPC(NF,MF,NC,MC,LEVELS,NM,STARTL,
     +          LENX,LENY,NT,TAU,REY,NCYCLE,NY1,NY2,TOL,
     +          U,V,P,IFAIL,
     +          WORK1,WORK2,WORK3,
     +          UN,UE,US,UW,VN,VE,VS,VW)
C
       T1 = TIMING(T1) - T0
       WRITE (4,'(//'' COMPUTATION TIME          : '',F8.2,
     +          '' SECS. (ON THE ALLIANT FX/4)''/)') T1
       WRITE (4,'('' IFAIL                       : '',I8)') IFAIL

       IF ( IFAIL .EQ. 0 ) THEN
       WRITE (4,'('' CONVERGENCE REACHED WITHIN'',I3,'' ITERATIONS'')')
     +      NCYCLE
       ELSE
       WRITE (4,'('' MAXIT ITERATIONS PERFORMED WITHOUT REACHING TOL'')')
       ENDIF

       END

       REAL FUNCTION UW(X,Y,T)
       REAL X,Y,T
         UW=0.
       RETURN
       END

       REAL FUNCTION UE(X,Y,T)
       REAL X,Y,T
         UE=0.
       RETURN
       END

       REAL FUNCTION VW(X,Y,T)
       REAL X,Y,T
         VW=0.
       RETURN
       END

       REAL FUNCTION VE(X,Y,T)
       REAL X,Y,T
         VE=0.
       RETURN
       END

       REAL FUNCTION US(X,Y,T)
```

```
REAL X,Y,T
  US=0.
RETURN
END

REAL FUNCTION UN(X,Y,T)
REAL X,Y,T
  UN=1.
RETURN
END

REAL FUNCTION VS(X,Y,T)
REAL X,Y,T
  VS=0.
RETURN
END

REAL FUNCTION VN(X,Y,T)
REAL X,Y,T
  VN=0.
RETURN
END

REAL FUNCTION TIMING()
REAL*4 TARRAY(2),ETIME

TIMING = ETIME(TARRAY)
TIMING = TARRAY(1)

RETURN
END
```

13.2. Program data

```
128
0.0078125
1000.0
20
2
1
2
0.0001
```

13.3. Program results

```
INTERVAL X-DIRECTION  [0,  :   1.0000
INTERVAL Y-DIRECTION  [0,  :   1.0000
MESH-WIDTH IN X-DIRECTION  :   0.0078
MESH-WIDTH IN Y-DIRECTION  :   0.0078
REYNOLDS-NUMBER            :   1000.0
```

```
TIME-STEP                 :   0.0078
NUMBER OF TIME-STEPS      :      128
NUMBER OF MG-CYCLES       :       20
START LEVEL               :        2
TOL                       :   .1E-03
ITERATIONS BEFORE CGC     :        2
ITERATIONS AFTER CGC      :        1


COMPUTATION TIME          :   133.53 SECS. (ON THE ALLIANT FX/4)

IFAIL                     :        0
CONVERGENCE REACHED WITHIN 20 ITERATIONS
```