M4D Named Arrays

M4D includes an array handling system so that arrays needed by more than one routine are available by name. The user may access the arrays of type double, integer or character at the command (input file) level to print or edit these arrays. The user may also create arrays, which can be useful for post processing. See, e.g., commands c_algebra, c_edit, c_print, c_prints, c_constant, and c_varinit. Arrays of type pointer are not accessible at the command level except that coefficient arrays (name style xx_c, xx_i, xx_n) may be printed using c_printscoef. In the below list of fixed array names, the arrays are color coded by type:

double, integer, character, pointer.

The size of most of the arrays relate to the grid dimensions. The size of the array will be designated in [ ]. With id, jd, kd as the dimensions of the 3-d grid[1]:

[g] on the grid-point arrays, size=id*jd*kd

[m] between the point (mid-point) arrays, size=(id-1)*(jd-1)*kd-1)

[p] pressure-grid arrays, size=(id+1)*(jd+1)*(kd+1)

[d] double grid arrays, size=(2*id-1)*(2*jd-1)*(2*kd-1)

Note that arrays are stored as arrays with only 1 dimension. (Arrays of pointers, may be considered to have 2 dimensions). If the description gives multiple dimensions within the brackets, e,g, [id*jd], the array is dimensioned id*jd, and the values are stored with the first index incrementing first, i=0,j=0 then i=1,j=0, etc.

While M4D uses a single 3-d grid in space, it allows coincident points so the number of equations needed may be much smaller than the number of points. Some arrays are only defined over the needed points. These dimensions are designated by:

[G] the number of independent grid point equations

[P] the number of independent pressure-point equations

[W] the number of independent wall grid points

For arrays which are not proportional to these, or which don't have fixed dimensions the designation [o] (other) will be used in the list on the next page with a more complete description to follow below.

Do not use the names in the list for user created arrays unless you are sure there will be no conflict in meaning or size. Note that array names are case sensitive, e.g., *T* and *t* are different. Also do not use *c:* as an array name as this is the command indicator. The letter following the name and size means:

c - created by a command;

n - not created by a command but needed by a command;

f - neither created or needed but used by a command if it is available.

E - associated only with an experimental model routine

Prefix = known to need update of documentation.

---

[1] Most of M4D is dimensioned 4-d with the 4th dimension initially envisioned as time with dimension nd. However this has never implemented. Time dependent calculations are done as (iterative-implicit) time marching. I.e., nd=1, and the grid space is effectively 3-d.

Alphabetical list of fixed array names

abcd [o] c
abcddouble [o] c
aveiflow [o] c
averages [7*21] c
beqd [G*6*6] c
beqex [m*12] c
beqm [G*6] c
bij [g*6] n
bijm [m*6] c
blockijk? [o] f
cam [G] c
cama [G] c
camb [G] c
cambij [G] c
caomturb [G] c
caqturb [G] c
clt [g] c
cltdouble [g] c
cltp [p] c
coef_c [G] c
coef_i [G] c
coef_n [G] c
contar [P*3] c
cpc_c [P] c
cpc_i [P] c
cpc_n [P] c
cpc1_n [P] c
cpda [8*3*g] c
cpflop_c [G] c
cpflop_i [G] c
cpflop_n [G] c
cplus [G] c
cpsleep [p*2] c
cpsleepm [p*2] f
csym [12] n
cvdc [g*3] c
cvdcdouble [g*12] c
cvdcrms [5] c
dbij [g*6] c
ddtall [m*6] c
ddtmax [1] c
disp [g] c
dp [p] c
dpdx [1] f
dU1 [g] c
dU2 [g] c
dU3 [g] c
dU1dt [g] nE
dU2dt [g] nE
dU3dt [g] nE
elpd [m] cE
eqnerrp [6] c
flowinn [o] c
flowinprop [o] c
flowout [o] c
flowoutprop [o] c
gbij [m] c
idim4d [4] c
idim4ddouble [4] c
idim4dp [4] c
ijkcorner [o] c
info4print [20] c
ipeqexit [o] c
ipfix [4] f
ITER [1] f
itrange [2] f
Lddx2 [m] cE
Lles [m] cE
match [g] c
matchpc [p*2] c
matchside [p*6] c
nocoefs [1] c
nogpts [1] c
noindfwpts [o] c
noindppts [o] c
noindwpts [o] c
noppts [1] c
omturb [g] n
pg [g] n
pkdk [m] c
pose [g] c
pp [p] n
prlam [1] n
p.defaultdir [1] f
p.fnames [3] n
p.fparms [6] n
p.gnames [4] n
p.gparms [13] n
p,grange [11] n
p.gxx [6] n
p.rgbcolor [o] c
p.vnames [3] n
p.vparms [11] n
qmid [m] n
qnoise [1] n
qturb [g] n
ray [m] cE
Repr [9*g] c
rho [g] n
rhom [m] n
rhsbij [G*6] c
rhsc [P] c
rhsomturb [G] c
rhsqturb [G] c
rhsU1 [G] c
rhsU2 [G] c
rhsU3 [G] c
rmsmm [2*10] c
roundoff [1] f
roundoffd [1] f
sijbij [m] cE
srate [m] c
Stpr [9*m] c
sum [g] c
TIME [1] f
timespace [2] c
U1 [g] n
U2 [g] n
U3 [g] n
U1ddxi [m*3] f
U2ddxi [m*3] f
U3ddxi [m*3] f
updatew [3] c
vlam [m] n
vlamg [g] n
vcoakley [m] c
vles [m] cE
vmarv [m*6] c
vmarvheat [m] c
volcont [m] c
volmom [G] c
vqnoise [m] c
vtpd [m] cE
Vvec [3*m] c
w2wdist [P] cE
w2wline [P*3] cE
w2wnorm [P*3] cE
walldist [P] c
walln2m [P*3] c
wherefw [g] c
wherep [p] c
whoelse [g] c
whoisfw [G] c
whoisp [P] c
whoisw [W] c
wnear [W] c
wnorm [3*W] c
wrate [m] nE
xyz [g*3] c
xyzdouble [d*3] c
xyzp [p*3] c
xyzpold [p*3] n
zrotation [1 or 3] f

Description of Named Array

Name [size] (type) At end: (m/y), last date the documentation was checked.

abcd [id+jd+kd+nd] (double) Set by c_gridfrommefp.
1-D interpolation parameters for each of the grid dimensions, a[i], b[j], c[k], d[n]. Provides as easy way to specify a grid plane or a point in the grid independent of the number of grid points.
Needed by c_areaflowint, c_aveijk, c_editabcd, c_gridcorner, c_gridtomefp, c_gridvarmod, c_lineoutput, ec_rayleigh, c_varinit, pc_abcmask, pc_xytocgrid, pc_xyzicut. (12/14)

abcddouble [2*id-1+2*jd-1+2*kd-1+nd] (double) Set by pc_gridinfo.
1-D interpolation parameters like *abcd,* but corresponding to the double grid, *xyzdouble*. Between the grid points values for adouble[i] are simple averages of a[i]. Available for use so that *xyzdouble* may be plotted.
Not used by other routines. (4/15)

aveiflow [(noprop+1)*iplanes] (double) Set by pc_iplaneint.c
Area, then area average of each property, for each of the plot iplanes. See command pc_iplaneint.
Not used by other routines. (12/14)

averages [7*21] (double) Set by c_areaflowint.
Area and mass averages set by command c_areaflowint for a specified portion of a grid plane. The first 7 values (*averages*[0] through *averages*[6]) are the magnitude of the area (sum of all the magnitude of all the area segments), the x, then y, then z components of the area vector, the positive flow through the surface (+*L* direction), the negative flow through the surface, and the net flow through the surface. The next 7 items (*averages*[7] – *averages*[13]) are corresponding averages (area average etc.) for the first property that was specified in the call to c_areaflowint, these are followed by the averages for the next property, etc. Note that the area or mass integral is returned instead of the average if *averages*[0] (through *averages*[6]) is zero.
Not used by other routines. (12/14)

beqd [G*6*6] (double) Set by c_bijrhsmarv(s), ec_bijrhsmarvex.
beqm [G*6] (double)
The source term for the six b_{ij} equations has an explicit part, *beqm*, and an implicit part evaluated in terms of the (center-point) change in b_{ij} and coefficients, *beqd*,
$\int \rho S_{b_{ij}} dVol = beqm_{ij} + beqd_{ij,mn} \Delta b_{mn}$. (For ij = 1,,6; implied sum mn = 1,,6.)
Needed by c_eqnsolvebij. (12/14)

beqex [m*12] (double) Set by c_bijrhsmarv(s) or ec_bijrhsmarvex.
Between-the-point values of intermediate Reynolds stress model parameters. Recompile with bug=0 (see source code) to omit saving these parameters which are not used by other routines. As of 12/14 the parameters are: $P/\tilde{\varepsilon}$, $\tilde{R}_t$, g_{eff}, f_w, f_b, c_1, c_3, c_4, c_6, f_r, G, $c_{1,\text{homo}}$.

bij [g*6] (double) Initialized by the user.
The 6 components, b_{11}, b_{22}, b_{33}, b_{12}, b_{13}, b_{23} of the anisotropy of the Reynolds stress tensor for the MARV family of Reynolds stress models.
Needed by c_bijrhsmarv(s), c_bijwallmarv, c_bijwallsplat c_eqnsolvebij, c_set_gbij, c_viscmarv, ec_bijrhsmarvex. Used if specified by c_momrhsr. Used if available by c_prinstress. (12/14)

bijm [m*6] (double) Set by ec_qbijles.
The between-the-points values of the 6 components, b_{11}, b_{22}, b_{33}, b_{12}, b_{13}, b_{23}, of the anisotropy of the Reynolds stress tensor. Set by ec_qbijles for a non-Bousssinesq LES model being tested.
Used if specified by c_momrhsr. (12/14)

blockijk? [various] (integer) Set by c_blocksetabc.
Grid index information for a block solve solution of the pressure-correction equations. "?" in the name is replaced by the letter specified in the input to c_blocksetabc. Several *blockijk?* arrays (each with a different final letter) may be set to create a multilevel block solve procedure.
Used if specified by c_eqnsolvep which calls routines located in blocksubs.c. (12/14)

cam [G] (double) Set by c_momcam.
Center point relaxation coefficients for dU1, dU2 and dU3 in the abbreviated momentum equations used in forming pressure correction equations.
Needed by c_contcpcdu, c_contdu. Used if available c_eqnupdatem. (12/14)

cama [G] (double) Set by c_momcamddt.
camb [G] (double)
Alternates to *cam*. Not used unless copied to *cam* (e.g. c: copy cama cam). For unsteady flows, *cama* appears to be a better choice that *cam* calculated by c_momcam. (12/14)

cambij [G] (double) Set by c_bijrhsmarv(s), ec_bijrhsmarvex.
Center point relaxation coefficients for the bij equations, based on *beqd*.
Needed by c_eqnsolvebij unless an alternative is specified by the input. (12/14)

caomturb [G] (double) Set by c_omrhsmarv or c_omrhscoakley.
Center point relaxation coefficients for *omturb* equations, based on the source term.

Designed to be used by c_eqnsolves. (Specify as an input parameter.) (12/14)

caqturb [G] (double) Set by c_qturbrhs.

Center point relaxation coefficients for *qturb* equations, based on the source term.

Designed to be used by c_eqnsolves. (Specify as an input parameter.) (12/14)

clt [g] (char) Set by c_gridfrommefp.

Single character point type array. w – wall, f – flow, s – internal solid. Other point types may be set by the user using c_edit to customize solution procedures, boundary conditions and control volume choices.

Needed by numerous routines. (12/14)

cltdouble [d] (char) Set by pc_gridinfo.

Like *clt* but corresponding to the double grid, *xyzdouble*. Between the points values are consistent with *clt,* but then the grid points themselves are set to 'x', so that on-the-grid surfaces plots of *xyzdouble* will show the outlines of the control volumes.

Not used by other routines. (4/15)

cltp [p] (char) Set by c_set_wherep.

Single character point types for pressure points, f – flow, F – flow next to a wall, s – internal solid, S – internal solid next to a wall. Points of type f may be modified using c_edit. c_eqnsolvep will omit solving for dp at points for which *cltp* ='o'. However do not modify points of type F, s, or S, as these indicators are used by other routines to determine the volume type.

Needed by numerous routines. (12/14)

coef_n [G] (int) Create with c_coefinit.

coef_i [G] (int *)

coef_c [G] (double *)

Coefficient arrays for the calculation of properties on the grid points. For the n^{th} independent point, *coef_n*[n] is the number of points in the coefficient set, *coef_i*[n][i] gives the independent point indices for the points, and *coef_c*[n][i+k**coef_n*[n]] are the coefficients for the i^{th} point and the k^{th} set of coefficients. For example an equation for a conserved species, s, at the n^{th} independent point might be written as

$$\sum_{i=0}^{i<coef\_n[n]} coef\_c[n][i]*s[coef\_i[n][i]]=0.$$

and the user may store contributions separately, e.g. the k^{th} set of coefficients may be set so that $\sum_{i=0}^{i<coef\_n[n]} coef\_c[n][i]*s[coef\_i[n][i+k*coef\_n[n]]]$ represents the contribution from convection. Create arrays coef_n, coef_i, and

coef_c using the command c_coefinit which also sets *nocoefs*[0], the maximum number of coefficient sets. The number of elements in the second dimensions for *coef_i* and *coef_c* are set (and reset) dynamically for each independent grid point as the routines determine the maximum number of points needed for each coefficient set. I.e. *coef_i* and *coef_c* are 'ragged arrays'.

Set coefficients using c_coefconv(step) for convection, c_coefvisc(step) for viscous terms and c_coefdt for time term. Use c_coefadd to algebraically combine the terms. Modify for stability (if needed) using c_coeffix. Needed by c_bijrhsmarv(s,ex), c_coefcplus, c_coefrhs, c_momcam(ddt), c_eqnsolvebij, c_eqnsolves, c_eqnupdatem. (12/14)

contar [P*3] (double) Set by c_set_contar.

The aspect ratios of continuity control volumes. Set for each continuity control volume which contains an independent p-point. Set as one over the perpendicular width of the control volume in each of the i,j,k grid directions, times the smallest of the 3 widths. E.g. if the grid is Cartesian with spacing $\delta x,\delta y,\delta z$=10,1,2, the *contar* values for that volume are 0.1, 1, 0.5.

Used by c_set_cpda, which call c_set_contar if *contar* does not exist. (12/14)

cpc_n [P] (int) Set by c_contcpcdu

cpc_i [P] (int *)

cpc_c [P] (double *)

Coefficient arrays for the pressure correction equations used to satisfy continuity and pressure boundary conditions. For the n^{th} independent pressure point, *cpc_n*[n] is the number of pressure points in the coefficient set, *cpc_i*[n][i] gives the independent point indices for the points, and *cpc_c*[n][i+k**cpc_n*[n]] are the coefficients for the i^{th} point and the k^{th} set of coefficients. There are 2 sets of coefficients, i.e. k=0 or k=1.

Set using c_contcpcdu. Then modify for Rhie-Chow type changes for stability using c_contcpcfixed. c_contcpcinlet. Then use c_contcpxexit and/or c_contcpcinlet to add boundary equations if appropriate. Needed by c_contrhsp, c_eqnsolvep. (12/14)

cpc1_n [P] (int) Set by c_contcpcfixed

The number of coefficients in the first cpc set which have absolute values greater than *roundoff* (or 1.e-8) times the center point coefficient.

Used by c_eqnsolvep (if available) instead if cpc_n. (12/14)

cpda [2*2*2*3*g] (double) Set by c_set_cpda.

The contribution to the pressure integral for L^{th} momentum component at point i,j,k from the 8 surrounding pressure points is

$$\int \overline{P} dA_L = \sum_{ia,ja,ka=0,1} cpda[ia,ja,ka,L,i,j,k] * pp[i+ia,j+ja,k+ka] \ .$$

Set to zero for wall and solid points.

Needed by c_set_cpflop. (12/14)

cpflop_n [G] (int) Set by c_set_cpflop.
cpflop_i [G] (int *)
cpflop_c [G] (double *)
For the momentum control volume for each independent flow point n, the pressure integrals are

$$\int \bar{P} dA_L = \sum_{j=0}^{j<cpflop\_n[n]} cpflop\_c[n][j + L * cpflop\_n[n]] * pp[cpflop\_c[n][j]]$$

where L is indexed 0,1,2 for the x,y,z components..
Needed by c_contcpcdu, c_contcpcexit, c_contdu, and by c_momrhsr ('p' option). (12/14)

cplus [G] (double) Set by c_momcam(ddt).
The sum of the positive coefficients for the first set of coefficients in coef_c. The coefficients of dUL (dU1, dU2, dU3) in the abbreviated momentum equations for the n^{th} independent point are cplus[n]+cam[n].
Needed by c_contcpcdu, c_contcpcext, c_contdu. (12/14)

cpsleep [p*2] (double) Set by c_set_cpsleep.
The 2-point interpolation coefficients for sleeping pressure points,
pp[i,j,k,n] = *cpsleep*[i,j,k,n,0] * *pp*[*matchpc*[i,j,k,n,0]]
+ *cpsleep*[i,j,k,n,1] * *pp*[matchpc[i,j,k,n,1]].
Needed by c_eqnsolvep, c_interp_gtop, c_ppreset, c_set_cpflop. May be needed by c_eqppts2ppts. (12/14)

cpsleepm [p*2] (double) Set by user.
The 2-point interpolation coefficients for sleeping pressure points for centered control volumes. If wanted, set once using
c: cvdcinit .5 0 c: set_cpsleep c: copy cpsleep cpsleepm
Used if available by c_gradprop. (12/14)

csym [12] (char) Set by the user.
Specifies symmetry plane information. The first six characters describe the possible symmetry planes at the first and last i-grid planes, j-planes then k-planes for momentum and bij equations. Characters: n –no, x, y or z, – plane of symmetry of uniform x, y or z. The second six characters are to specify repeating boundaries and pressure symmetry. Characters: n–none, R – repeat with 2 grid points, r–repeat with more than 2 grid points, s–symmetry plane. Set by the user, e.g. using c: constant csym c 12 n n n n z z R R s s s s
Needed by numerous routines. (12/14)

cvdc [g*3] (double) Initialize with c_cvdcinit; reset with c_cvdcreset.
The fractional spacing parameters in the i,j,k directions (L=0,1,2) dividing up the continuity control volumes to form control volumes for properties on the

grid points. (The array is slightly over dimensioned for convenience.) Values may vary between 0 and 1. Reset the governing parameters using c_cvdcparm. Modify based on convection, and optionally viscosity and time step, using c_cvdcreset.

Needed by c_interp_gtop, and (indirectly through geom8 routines) by several other routines. (12/14)

cvdcdouble [(g*4)*3] (double) Initialize with c_cvdcinit; reset with c_cvdcreset.

The half grid spacing (so that lines and faces are included) breakup of the continuity control volumes. Used as 3 separate arrays, the breakup in the i-direction, cvdci[2*jd,2*kd,id], the j-direction, cvdj[2*kd,2*id,jd], and the k-direction, cvdk[2*id,2*jd,kd].

Needed by several routines (indirectly through geom8 routines). (12/14)

cvdcrms [5] (double) Set by c_cvdcreset

Five value analysis when control volume boundaries are reset. The first 3 values give the rms change (over the entire grid) in *cvdc* in the i, j, or k grid directions. The 4th value is the maximum change, with the 5th value being 0, 1, or 2 specifying the i, j, or k direction of the maximum.

Not used by other routines. (12/14)

dbij [g*6] (double) Set by c_eqnsolvebij.

The change in *bij* from solving the b_{ij} equations. Use c_algebra or (better) c_varupdate to update *bij* with the change, *dbij*. (12/14)

ddtall [m*6] (double) Set by c_ddtall.

Six 1/dt timescales set for the continuity control volumes.

n=0: convection - CFL number, e.g. for Cartesian, $\max(U_1/\delta x, U_2/\delta y, U_3/\delta z)$.

n=1: laminar viscosity, if *vlam* exists, e.g. for Cartesian, $\frac{vlam}{\rho}\left(\frac{1}{\delta x^2}+\frac{1}{\delta y^2}+\frac{1}{\delta z^2}\right)$.

n=2: $\max\limits_{i=1,3}\sqrt{\max\left(0,\sum\limits_{k=1}^{3}\left(\frac{\partial U_i}{\partial x_k}+2\varepsilon_{ink}\Omega_n\right)\left(\frac{\partial U_k}{\partial x_i}+2\varepsilon_{kni}\Omega_n\right)\right)}$ an alternate 'safe' 1/dt.

n=3 absolute vorticity, $\sqrt{2W_{ij}W_{ij}}$

n=4 strain rate, $\sqrt{2S_{ij}S_{ij}}$

n=5 $\max\limits_{i=1,3}\sqrt{\sum\limits_{k=1}^{3}\left|\left(\frac{\partial U_i}{\partial x_k}+2\varepsilon_{ink}\Omega_n\right)\left(\frac{\partial U_k}{\partial x_i}+2\varepsilon_{kni}\Omega_n\right)\right|}$ a center-point relaxation 1/dt useful for converging steady separated flow.

Not used by other routines but individual components may be useful for setting spatially non-uniform time steps when strict time-accuracy is not practical. (12/14)

ddtmax [1] (double) Set by c_ddtall

Maximum 1/dt based on convection (CFL). Invert it for uniform time step based on the CFL condition.
Not used by other routines. (12/14)

disp [g] (double) Set by c_coeffix.
The dispersion parameter for each independent grid point for the specified coefficient set. It is evaluated as the largest negative coefficient divided by the center point coefficient (values of *disp* are negative). If the center point coefficient is less than or equal to zero, *disp* = –10, if there is no equation for the point disp=0.
Not used by other routines. (12/14)

dp [p] (double) Set by c_eqnsolvep.
The change in pressure from solving the pressure correction equations.
Needed by c_contdu to update the velocities for continuity using the abbreviated momentum equations. Add to the current pressure solution with
c: algebra pp "" pp dp 1. 0. 1. (12/14)

dpdx [1] (double) Set by the user.
A global dp/dx which will be added to x-momentum if specified by c_momrhsr input parameters. Set, e.g., by c: constant dpdx d 1 .02 .
Used by c_momrhsr if exists and requested. (12/14)

dU1 [g] (double) Optionally set by c_eqnupdatem.
dU2 [g] (double)
dU3 [g] (double)
The change in the velocity components based on momentum which c_eqnupdatem has added to *U1, U2, and U3*.
Updated by c_contdu if available. Not used by other routines. (12/14)

dU1dt [g] (double) Set by user.
dU2dt [g] (double)
dU3dt [g] (double)
The unsteady time derivative of the velocity components at the grid points.
Needed by ec_qbijles, to set *bijm,* for a test non-Boussinesq LES model. (12/14)

elpd [m] (double) Set by ec_rayleigh.
The length scale for a mixing-length Rayleigh instability model.
Not used by other routines. (12/14)

eqnerrp [6] (double) Set by c_eqnsolvep.
Analysis of the effectiveness of the pressure equation solution procedure. First 2 values, the rms and maximum error in p before solution. Next 2

values, the rms and maximum error after solution. Last 2 values, the ratio of the after to before values.
Not used by other routines. (12/14)

flowinn [1+5*flowinn[0]] (int) Set by c_inletinit.
Parameters specifying the location of the inlet boundary regions: flowinn[0] = number of regions; for region k: flowinn[1+5*k] = number of grid points in the region, flowin[2+5*k] = index of the first point, flowin[3+5*k] = index of the last point, flowin[4+5*k] = inlet direction, (0,1,2 for i,j,k), flowin[5+5*k] = 0 if flow points are on the +i,j or k side of the inlet, or = –1 if on the negative side.
Used if available by c_contcpcinlet, c_inletreset. (12/14)

flowinprop [ntinlet,4] (double) Set by c_inletinit.
where ntinlet = total number of inlet grid points.
Parameters for the inlet boundary condition. For inlet boundary point k, flowinprop[k] = reference total pressure, flowprop[k+L*ntinlet] = ratio of velocity component UL (L=1,2,3) to the potential flow velocity for the reference total pressure.
Needed by c_contcpcinlet, c_inletreset if *flowinn* exists. (12/14)

flowout [1+16*flowout[0]] (int) Set by c_exitinit.
Parameters specifying the location of the outlet pressure boundary regions: flowout[0] = number of regions; then for each region: type, dL, ist[4], iend[4], ifix[4] (padded to 16 items). type=1: fixed p at ifix, dp/diL=uniform; type=2 (not coded as of Dec 2014) fixed pave, dp/diL=uniform; type=3 fixed flow rate, dp/diL=uniform, type 4 fixed flow rate, dp uniform. dL = + or – 1,2,3 indicating the normal grid direction pointing towards internal points. ist and iend are the first and last pressure grid indices for the region. ifix is a point on the pressure boundary at which the pressure is fixed (for type 1) and relative to which the other exit boundary equations are written.
Used by c_contcpcexit, c_contrhsexit if it exists. (12/14)

flowoutprop [flowout[0]] (double) Set by c_exitinit
Exit boundary condition parameter for each exit region: type=1, fixed pressure; type=2, fixed average pressure, type = 3 or 4, fixed mass flow rate. Note only types 1, 3 and 4 coded as of Dec 2014.
Needed by c_contcpcexit, c_contrhsexit if *flowout* exists. (12/14)

gbij [m] (double) Set by c_bijrhsmarv(s), c_set_gbij, ec_bijrhsmarvex.
Between the points values of the anisotropy parameter, *g*.
Needed by c_omrhsmarv. (12/14)

idim4d [4] (int) Set by c_gridfrommefp.

The 4d grid dimensions, 3 space, and time. Note, so far only a time dimension of 1 has been used.
Needed by numerous routines. (12/14)

idim4ddouble [4] (int) Set by pc_gridinfo.
The 4d grid dimensions, 3 space, and time of *xyzdouble*. Available for use so that *xyzdouble* may be plotted.
Not used by other routines. (4/15)

idim4dp [4] (int) Set by c_gridmatch.
The 4d grid dimensions, 3 space, and time of *xyzp*.
Not used by other routines. (4/15)

ijkcorner [9*number of corners+1] (int) Set by c_gridcorner.
Information about O-grid corners created by c_gridcorner. If the first of the nine values is >0, the next 4 are the grid indices starting the corner, and the following 4 the grid indices where the corner ends. A first value of 0, ends the list.
Not used by other routines.

info4print [20] (char *) Created by main program (m4d.c)
Used for print output management. See command printcontrol.
Used in main and by subroutines arrayhowto and printout. (4/15)

Modified by command printcontrol (part of m4d.c)
ipeqexit [1+ipeqexit[0]] (int) Set by c_contcpcexit.
The number of, then the pressure equation indices for the pressure equations associated with the exit boundary.
Used by c_eqnsolvep if available. (12/14)

ipfix [4] (int) Set by user.
The pressure-grid indices i,j,k,n (C-style, first value 0) for a point at which the change in pressure will be zero. Appropriate to set for configurations which use a repeating inflow/outflow boundary so the pressure level does not wander.
Used by c_eqnsolvep if available. (12/14)

ITER [1] (int) Optionally created by the user.
An iteration parameter which may be created by the user using, e.g. c: constant ITER I 1 0, and indexed, e,g,, by c: edit ITER i 1 m 1 add 1 1 1 end
If it exists, it is printed with the analysis done by c_areaflowint, c_rmsminmax and c_cvdcreset. (12/14)

itrange [2] (int) Optionally set by the user.

If set, computations are limited to the time index range from itrange[0] through itrange[1]. Note: to date (Dec 2014) values other than 0,0 have not been tested since dimensions other than 1 for the time index have not been used.

Totally unnecessary but used by numerous routines if available. (12/14)

Lddx2 [m] (double) Set by ec_qbijles

Lles [m] (double) Set by ec_viscles

Two length scales being test for LES. Set for each non-zero between-the points volume.

Lddx2 [j]= $\sqrt{3/(1/\delta x^2 + 1/\delta y^2 + 1/\delta z^2)}$ (Cartesian)

A length scale set as the smaller of the between the points array, *namecl*, and a VanDriest correction,

$$Lles[j] = \min(\, namecl[j],\ 0.41y\left(1-\exp\left(\frac{-y\sqrt{(\nu_{lam}+\nu_{LES})S}}{26\nu_{lam}}\right)\right).$$

Not used by any other routines. (12/14)

match [g] (int) Set by c_gridmatch.

match[m] equals the index of the primary point whose property values are the same as those for point m. Matches occur both for grid points which have identical (to within roundoff) locations and across repeating boundaries.

Needed by several routines. (12/14)

matchpc [p*2] (int) Set by c_gridmatch.

Two values for each pressure grid point, matchpc[m] and matchpc[m+noppts[0]]. For sleeping pressure points, matchpc gives the indices of the pressure points used for interpolating for the pressure value at the sleeping pressure point. The first value is –1 if the pressure point is not sleeping. For non-boundary pressure points, (i=1,,id-1, etc.) matchpc[m]>=0 indicates that the corresponding continuity control volume is sleeping (has no volume).

Needed by c_contcpcdu, c_cvdcinit, c_cvdcreset, c_ddtall, c_eqnsolvep, c_eqppts2ppts, c_exitinit, c_interp_gtop, c_ppreset, c_set_cpflop, c_set_cpsleep, c_set_volmom, c_set_wherep. May also be needed by c_gradprop. (12/14)

matchside [p*6] (char) Set by c_gridmatch.

For each of the 6 sides surrounding each pressure point (i-, i+, j-, j+, k-, k+), matchside gives a letter indicating whether and which other side is a match for this side. matchside = 'n', for none, I for +i side, i for –i side, J for +j side, j, for –j side, K for +k side, k for –k side, R for +side repeat, and r for –side repeat.

Needed by c_ cvdcinit, c_cvdcreset. (12/14)

nocoefs [1] (int) Set by c_coefinit.
The number of coefficient sets in coef_c.
Needed by c_coefadd, c_coefconv(step), c_coefcplus, c_coefdt, c_coeffix, c_coefrhs, c_coefvisc(step), c_coefzero, c_momrhsr. (12/14)

nogpts [1] (int) Set by c_gridmatch.
The number of grid points = idim4d[0]*idim4d[1]*idim4d[2]*idim4d[3].
Needed by c_coefrhs, c_eqnsolvebij, c_eqnsolves, c_eqnupdatem, c_varmatch. (12/14)

noindfwpts [1+nd+1] (int) Set by c_set_wherefw.
The first and last values are the number of independent grid points for which (property) equations may be needed. For time index, n, the associated equations are from noindfwpts[1+n] to noindfwpts[1+n+1].
Needed by numerous routines. (11/14)

noindppts[1+nd+1] (int) Set by c_set_wherep.
The first and last values are the number of independent pressure points for which equations may be needed. For time index, n, the associated equations are from noindppts[1+n] to noindppts[1+n+1].
Needed by numerous routines. (12/14)

noindwpts [1+nd+1] (int) Set by c_**wallnorm**.
The first and last values are the number of independent wall grid points. For time index, n, the associated wall points are from noindwpts[1+n] to noindwpts[1+n+1].
Needed by c_bijwallmarv, c_bijwallsplat, c_omwall, c_omwallcoakley, c_omwallmarv(s). (12/14)

noppts [1] (int) Set by c_gridmatch.
The number of pressure grid points = (idim4d[0]+1)* (idim4d[1]+1)* (idim4d[2]+1)* idim4d[3].
Used by c_eqnsolvep, c_eqppts2ppts, c_interp_gtop, c_ppreset. May also be needed by c_gradprop. (12/14)

omturb [g] (double) Initialized by the user.
The specific dissipation rate of the turbulence, $\omega = \tilde{\varepsilon}/k$, a turbulence variable in both the MARV Reynolds stress model and the Coakley 2-Eq. Model. Note that when between-the-points values for ω are needed, averages of log(ω) may used instead of the linear averages which are in general used for other properties. Update after solving ω equations with c: varupdate omturb nameofdom log (where nameofdom is the name for the change in omturb) to maintain values >=0. If zero values become a problem, maintain a minimum value using c: function omturb max omturb omturbmin (where omturbmin is a constant set using c_constant.)

Set wall values using c_omwall, c_omwallcoakley, or c_omwallmarv(s).
Needed by c_bijrhsmarv(s), c_bijwallmarv, c_omrhscoakley, c_omrhsmarv, c_qturbrhs, c_visccoakley, c_viscmarv, c_viscmarvheat, c_viscnoise, ec_bijrhsmarvex. (12/14)

pg [g] (double) Set by the user.
Static pressure on the grid points. May be set from *pp* using c: interp_ptog 1 pp pg .
May be needed by c_inletinit. (12/14)

pkdk [m] (double) Set by c_bijrhsmarv(s), c_set_pkdk, ec_bijrhsmarvex.
Between the points values of the turbulence production rate divided by the turbulence kinetic energy.
Needed by c_omrhscoakley, c_omrhsmarv and c_qturbrhs. (12/14)

pose [g] (double) Set by c_coeffix.
For the specified coefficient set pose is evaluated as the center point coefficient divided by the sum of the positive coefficients. If the sum of the positive coefficients is less than or equal to zero, pose= –10, if there is no equation for the point pose=1.
Not used by other routines. (12/14)

pp [noppts] (double) Initialized by the user.
Static pressure variable on the pressure grid. For incompressible flow procedures pressure is the variable which enforces continuity by setting up and solving pressure-correction equations.
Modified by c_inletreset, c_ppreset. Also needed by c_contcpcinlet, c_contrhsexit, c_contrhsp. Use as in input parameter for c_momrhsr to add the pressure gradient term to the momentum r.h.s, (12/14)

prlam [1] (double) Initialized by the user.
The laminar Prandtl number
Needed by c_viscmarvheat. (12/14)

p.??? Arrays starting with p. are plot package arrays, mostly for specifying input.

p.defaultdir [1 or more] (char *) Set by user.
For the plot package to output .gif files it needs the file color.map, and to label plots it needs the files font.list and letters18x30.fmap. If these are not in the current directory it searches the directories listed in *p.defaultdir* for them.

Example: c: constant p.defaultdir s 1 Users/Shared/a/jgm

Used if needed and available by pc_lineplot, pc_bar, pc_image. (4/15)

p.fnames [3] (char *) Set by user.

Names for color-fill plotting: *property cltrange nyout*.
property - name of on-, between- or p-points property to be contoured;
cltrange - the *clt* values over which the contours are filled;
nyout - if the first letter is 'n' contours are limited to the range in p.fparms, if not the end colors are used to fill outside values.

Example: c: constant p.fnames s 3 qturb fi nout

Needed by pc_bar, and by pc_picture if color-fill is specified. (4/15)

p.fparms [6] (double) Set by user.

Color-fill parameters: *propstart propfix propend colorstart colorfix coloreend.*
The normal colorfill range associates *propstart* with *colorstart* and *propend* with *colorend* and uses a linear variation between these. If *propstart=propend* the program determines the range from the min, max values of the property. If *propstart* < *propfix* < *propend* and colorfix >=0, *propfix* is associated with *colorfix* and a two-part color bar range is used for color-fill. Set *propend* >= *propstart* but *colorend* may be > or < *colorstart*. The standard color.map file gives a sequence of colors from index 11 thru 50 appropriate for color-filling.

Example: c: constant p.fparms d 6 -1 0. 3 50 -1 11

Needed by pc_bar, and by pc_picture if color-fill is specified. (4/15)

p.gnames [4] (char *) Set by user.

Names of the grid arrays. These may be for the on-the-points grid *xyz*, the *xyzdouble* grid, or a grid created using pc_xytocgrid or pc_xyzicut. (Note: pc_picture requires that color-fill properties or velocity components must correspond in dimensions to the specified grid.)

Examples:

c: constant p.gnames s 4 idim4d xyz abcd clt
c: constant p.gnames s 4 idim4ddouble xyzdouble abcddouble cltdouble

Needed by pc_picture, pc_iplaneint. (4/15)

p.gparms [13] (int) Set by user.

Parameters for plotting grid lines: *lw if iw is io jf jw js jo kf kw ks ko*.
lw - line width in pixels;
if iw is io - color numbers for i-grid lines, for *clt* types f, w, s, or other, respectively;
jf jw js jo kf kw ks ko - similarly for j and k grid lines.

Example: c: constant p.gparms i 13 2 0 1 0 0 0 1 0 0 0 1 0 0

Needed by pc_picture if grid lines are requested. (4/15)

p,grange [11] (double) Set by user.

Plot grid range and anchor: *as ae bs be cs ce apt bpt cpt facross fup.*
pc-picture limits the scope of the plot based on the 1-d grid parameters, *as<=a<=ae*, *bs<=b<=be, cs<=c<=ce.* The image is anchored by point *apt, bpt, cpt* being located at the fractional distance *facross* from left to right on the image and at a fractional height *fup* from bottom to top. Note that the plot may also be limited by the chosen scaling and size of the image (See command pc_picture)

Example: c: constant p.grange d 11 -2 1. 0 2 0 0 -2 0 0 .02 .02

Needed by pc_picture. (4/15)

p.gxx [6] (double) Set by user.
Cartesian across and up vectors for the plot:
xacross yacross zacross xup yup zup.

Example: c: constant p.gxx d 6 1 0 0 0 1 0

Needed by pc_picture. (4/15)

p.rgbcolor [3*number_of_colors] (char) Set by p_outgif called by pc_image.
The red, green, blue intensities for each color in the color map file, color.map. Integer values are from 0 to 255, so they are saved as characters. Note, consistent with the creation of .gif files, the color map is limited to 256 colors.
Set and used by the outgif option of pc_image. (4/15)

p.vnames [3] (char *) Set by user.
Names of the x,y,z velocity components for plotting velocity vectors:

Example: c: constant p.vnames s 3 U1 U2 U3

Needed by pc_iplaneint and the vector option of pc_picture.

p.vparms [11] (double)
Parameters for plotting velocity vectors: *dt headmax headf nybet lwpixel colotz colors colore dUz Uns Une.*
The component of the velocity vector in the viewing plane is shown with length *dt* times its magnitude (the result in geometry units). The head for the velocity vector is the smaller of *headmax* (in geometry units) and *headf* times the length of the vector. If *nybet* = 0, on-the-points vectors are shown, otherwise between-the-points vectors are shown (in directions where more than 1 plane is included in the plot range). The linewidth for the vectors are *lwpixel*, in pixels.
The out-of-plane velocity component is shown by color. Color index *colortz* is used when the magnitude of out of plane component is < *dUz.* Otherwise a linear variation is used relating of color index with out of plane velocity, *U* from *colors* for *U<= Uns* to *colore* for *U>= Une.*

Examaple: c: constant p.vparms d 11 2. .04 .2 0 1 5 11 20 0.2 -3. 3.

Needed by the velocity vector option of pc_picture. (4/15)

qmid [m] (double)
Between-the-points values for *qturb* to be used with *bijm*, the between-the-points values of b_{ij}. May be set as an effective *qturb* for LES.
Used if available by c_momrhsr, for the *bijm* option. (12/14)

qnoise [1] (double) Initialized by the user.
The velocity scale for the noise turbulence
Needed by c_viscnoise. (12/14)

qturb [g] (double) Initialized by the user.
The square root of the turbulence kinetic energy. A turbulence variable for both the MARV Reynolds stress model and the Coakley 2=Eq Model.
Needed by c_bijrhsmarv(s), c_omrhscoakley, c_omrhsmarv, c_omwall, c_omwallcoakley, c_omwallmarv(s), c_qturbrhs, c_set_pkdk c_visccoakley, c_viscmarv, c_viscmarvheat, c_viscnoise, ec_bijrhsmarvex. Used if available by c_prinstress and the *bij* option of c_momrhsr. (12/14)

ray [m] (double) Set by ec_rayleigh
Instability indicator. Set to 1 in regions where a peak in vorticity not to close to a wall suggests the flow may have a Helmholtz instability. Otherwise = 0.
Not used by other routines. (12/14)

Repr [3*3*g] (double) Set by c_prinstress.
The principle vectors for the Reynolds stress tensor. Set by c_prinstress if *bij* and *qturb* exist.
Not used by other routines. Can be dumped and then used by the JGM program x11pictw. (12/14)

rho [g] (double) Set by the user.
Fluid density on the grid points.
Used by numerous routines. (12/14)

rhom [m] (double)
Fluid density between the grid points.
Used if available by c_momrhsr, for the *bijm* option. (12/14)

rhsbij [G*6] (double) Set by c_bijrhsmarv(s), ec_bijrhsmarvex.
Right hand side for the bij equations.
Needed by c_eqnsolvebij. (12/14)

rhsc [P] (double) Set by c_contrhsu.

Right hand side of the continuity (pressure correction) equations. Set by c_contrhsu as $rhsc[]=-\oint \rho U_i dA_i$ for each continuity control volume.

Pressure stability corrections are added by c_contrhsp.c. Modified for the exit boundary condition by c_contrhsexit. Needed by c_eqnsolvep. (12/14)

rhsomturb [G] (double) Set by c_omrhscoakley or c_omrhsmarv.

Right hand side of the equations for *omturb*. Use with c_eqnsolves when solving the equations. (12/14)

rhsqturb [G] (double) Set by c_qturbrhs.

Right hand side of the equation for *qturb*. Use with c_eqnsolves when solving the equations. (12/14)

rhsU1 [G] (double) Set by c_momrhsr.
rhsU2 [G] (double)
rhsU3 [G] (double)

Right hand side of the x, y and z momentum equations.
Needed by c_eqnupdatem. (12/14)

rmsmm [2*10] (double) Set by c_rmsminmax

The root mean square (weighted simply by the number of values) and the largest magnitude for the first 10 variables analyzed by c_rmsminmax.
Not used by other routines. (12/14)

roundoff [1] (double) Optionally set by the user

If set by the user, it gives the level at which relative values of coefficients are considered negligible (and may be set to zero). It is also used to determine when grid points are considered coincident (relative value of difference compared with values to within *roundoff*). Set using c_constant.
Used if available by c_coefconv(step), c_coefdt, c_coeffix, c_coefvisc(step), c_coefzero, c_contcpcdu, c_contcpcexit, c_gridfrommefp, c_gridmatch, c_set_cpda, c_set_cpflop. (12/14)

roundoffd [1] (double) Optionally set by the user

If set by the user, it gives the level at which relative contributions to the r.h.s. of equations, or to the evaluation of gradients is negligible and may be set to zero.
Used if available c_coefrhs and c_gradprop. (12/14)

sijbij [m] (double) Set by ec_qbijles.

Between the points values of $S_{ij}b_{ij}$ for the LES model b_{ij} in ec_qbijles.
Not used by other routines. (12/14)

srate [m] (double) Set by ec_qbijles, c_set_srate, ec_viscles.
The magnitude of the strain-rate, $\sqrt{2S_{ij}S_{ij}}$. Note the velocity gradients are corrected for incompressible continuity ($S_{ii}=0$) before evaluation.
Not used by other routines. (12/14)

Stpr [3*3,m] (double) Set by c_prinstress.
The principle vectors for the relative velocity strain tensor. Set by c_prinstress if *U1ddxi, U2ddxi, U3ddxi* exist.
Not used by other routines. Can be dumped and then used by the JGM program x11pictw. (12/14)

sum [g] (double) Set by c_coeffix,
The sum of the coefficients in the specified coefficient set for each grid point.
Not used by other routines. (12/14)

TIME [1] (double) Set by the user.
The current time in time dependent calculations. It may be set using c_constant, and indexed using c_edit or c_algebra.
If it exists, it is printed with the analysis done by c_areaflowint, c_rmsminmax and c_cvdcreset. (12/14)

timespace [2] (double) Set by m4d.c (main program)
Updated between each command, *timespace*[0] is the total c.p.u. time used is seconds; *timespace*[1] is the maximum memory used in bytes.
Not used by any routines but may be printed or dumped. (12/14)

U1 [g] (double) Initialized by the user.
U2 [g] (double)
U3 [g] (double)
The x,y,z (relative in rotating frames) velocity components on the grid points. After setting up the momentum equations, update based on momentum using c_eqnupdatem. After solving the pressure correction equations update for continuity using c_contdu and for the inlet boundary c_inletreset.
Needed by numerous routines. (12/14)

U1ddxi [m*3] (double) Set using c_gradprop
U2ddxi [m*3] (double)
U3ddxi [m*,3] (double)
Velocity gradients in the x, y and z directions. Set using c: gradprop 1 U1 U2 U3 "" .
Used by c_prinstress if available. (12/14)

updatew [3] (double) Set by c_varupdate.

An array showing how much c_varupdate has restricted the changes applied to variable. With respect to a reduction factor F, i.e. *name*[j] = *name*[j]+F**change*[j], *updatew*[0] = F_{min}, *updatew*[1]= the number of points with F<0.5 and *updatew*[2] = the number of points with F<0.001. (For method "b" F represents an average factor.)

Not used by other routines. (4/15)

vlam [m] (double) Set by the user.

Between-the-points values of the laminar viscosity.

Needed by c_bijrhsmarv(s), c_omrhscoakley, c_omrhsmarv, c_visccoakley, ec_viscles, c_viscmarvheat, c_viscnoise, ec_bijrhsmarvex. Used if available by c_ddtall. Use with c_coefvisc to set laminar viscous coefficients. (12/14)

vlamg [g] (double) Set by the user.

On-the-points values of the laminar viscosity.

Needed by c_omwall, (12/14)

vcoakley [m] (double) Set by c_visccoakley.

Between-the-points values of the isotropic turbulent viscosity for the Coakley model.

Use with c_coefvisc to set turbulent viscous coefficients and c_set_pkdk to set *pkdk*.

Not used by other routines. (12/14)

vles [m] (double) Set by ec_viscles.

Between-the-points values of an isotropic turbulent viscosity based on an LES mixing length type model. (Not used as of Dec 2014.)

Not used by other routines. (12/14)

vmarv [m*6] (double) Set by c_viscmarv.

Between-the-points values of the nonisotropic turbulent viscosity, for turbulent diffusion of *bij* and *qturb* in the MARV Reynolds stress model.

Use with c_coefvisc to set turbulent viscous coefficients. For MARV model turbulent diffusion of *omturb*, multiply *vmarv* or the resulting viscous coefficients by 1.3666.

Not used by other routines. (12/14)

vmarvheat [m] (double) Set by c_viscmarvheat.

Between-the-points values of an isotropic turbulent viscosity for heat and mass transfer compatible with the MARV model.

Use with c_coefvisc to set turbulent viscous coefficients for heat or mass transfer.

Not used by other routines. (12/14)

volcont [m] (double) Set by c_set_volcont.

Between-the-points physical volume. I.e. volume of continuity control volumes.

Not used by other routines. (12/14)

volmom [G] (double) Set by c_set_volmom.
The physical volume associated with the equations for each independent grid point.
Needed by c_contcpcdu. (12/14)

vqnoise [m] (double) Set by c_viscqnoise.
Between-the-points values of an isotropic turbulent viscosity to approximate the effects of background noise.
Use with c_set_pkdk to determine extra noise turbulence production. Not used bu other routines. (12/14)

vtpd [m] (double) Set by ec_rayleigh.
Between-the-points values of an isotropic turbulenct viscosity to approximate pressure-diffusion of turbulence kinetic energy in regions of isotropic turbulence. Part of a test turbulence model for 2-d instabilities.
Not used by other routines. (12/14)

Vvec [3,m] (double) Set by c_prinstress.
The relative vorticity vector. Set by c_prinstress if *U1ddxi, U2ddxi, U3ddxi* exist.
Not used by other routines. (12/14)

w2wdist [P] (double) Set by ec_w2wlineflat.
w2wline [P*3] (double)
w2wnorn [P*3] (double)
Arrays calculated by a starter routine for a corner reflection model. Very geometrically limited.
Not used by other routines. (12/14)

walldist [P] (double) Set by c_walldist.
Distance to the nearest wall from the mid-point of each continuity control volumes. *walldist*[] is set to a negative value for locations which are not valid continuity control volumes.
Needed by c_omrhscoakley, ec_rayleigh, c_visccoakley, ec_viscles, c_viscmarvheat, and by c_bijrhsmarv(s) and ec_bijrhsmarvex for rotating walls if *zrotation* exists. (12/14)

walln2m [P,3] (double) Set by c_walldist.
Components of the wall-normal unit vector pointing to the mid-point of each continuity control volumes. Set only for valid continuity control volumes.
Not used by other routines, but a building block for wall reflection models. (12/14)

wherefw [g] (int) Set by c_set_wherefw.

The equation number associated with the grid point in the noindfwpts list. Set for matched points as well as primary points. wherefw[] = –1 if there is no equation associated with the grid point.

Needed by numerous routines. (12/14)

wherep [p] (int) Set by c_set_wherep.

The equation number for the pressure grid point in the noindppts list. wherep[] = –1 if not an independent point so that the value will be –1 for sleeping (zero volume) continuity control volumes.

Needed by numerous routines. (12/14)

whoelse [g] (int *) Set by c_gridmatch.

Array of integers for each grid point. For independent grid points, whoelse[m][0] gives the number of other points which match this grid point. whoelse[m][1] to whoelse[m][whoelse[m][0]] gives the indices of the multiple points. For non-independent grid points, whoesle[m]=0.

Needed by c_bijwallmarv(s,ex), c_bijwallsplat, c_interp_ptog, c_omwall, c_omwallcoakley, c_omwallmarv, c_set_wherefw. (12/14)

whoisfw [G] (int) Set by c_set_wherefw.

whoisfw[m] gives the primary grid point index for equation m in the noindfwpts list. Note: wherefw[whoisfw[m]]=m.

Needed by c_bijrhsmarv(s), c_coefconv(step), c_coefdt, c_coeffix, c_coefvisc(step), c_coefzero, c_eqnsolvebij, c_eqnsolves, c_eqnupdatem, c_eqpts2gpts, c_gpts2eqpts, c_set_cpflop, ec_bijrhsmarvex. (12/14)

whoisp [noindppts] (int) Set by c_set_wherep.

whoisp[m] gives the pressure grid point index for equation m in the noindppts list. Note: wherep[whoisp[m]]=m.

Needed by c_contcpcfixed, c_eqnsolvep, c_eqppts2ppts, c_ppts2eqppts, c_set_cpda, c_walldist. (12/14)

whoisw [noindwpts] (int) Set by c_wallnorm.

whoisw[m] gives the primary grid point index for the m'th wall point in the noindwpts list.

Needed by c_bijwallmarv, c_bijwallsplat, c_omwall, c_omwallcoakley, c_omwallmarv. (12/14)

wnear [W] (int) Set by c_wallnorm.

wnear[m] gives the index of the near wall grid point for the m'th wall point in the noindwpts list.

Needed by c_bijwallmarv, c_bijwallsplat, c_omwall, c_omwallcoakley, c_omwallmarv. (12/14)

wnorm [3*W] (double) Set by c_wallnorm.
The wall normal unit vectors for the points in the noindwpts list.
Needed by c_bijwallmarv, c_bijwallsplat, c_omwall, c_omwallcoakley, c_omwallmarv. (12/14)

wrate [m] (double) Set by user.
Between-the-points magnitude of the vorticity, $\sqrt{2W_{ij}W_{ij}}$. May be set by c: ddtall then c: alias ddtall 6 ddtcfl ddtlam ddtsafe wrate srate ddtold .
Needed by ec_rayleigh. (12/14)

xyz [g*3] (double) Set by c_gridfrommefp.
The x, y, z values for the grid points. Modified by c_gridcorner, c_gridvarmod, and possibly c_gridmatch.
Needed by numerous routines either directly or through calls to geomc or geom8 routines. (12/14)

xyzdouble [d,3] (double) Set by c_set_xyzdouble.
Nominal half spacing grid using *cvdc* and *cvdcdouble* to break up the continuity control volumes.
Needed by, c_coefvisc, c_interp_ptod, c_momrhs, c_set_cpda through geom8 routine. (12/14)

xyzp [p,3] (double) Set by c_set_xyzdouble.
The x,y,z locations of the pressure points. These are located where the continuity control volumes are divided up as well as points on the grid boundaries. Note xyzp is a subset of xyzdouble but is stored as a separate array for convenience.
Needed by c_contcpcdu, c_inletreset, c_interp_ptod, c_interp_ptog, c_ppreset, c_set_cpda, c_set_cpsleep. (12/14)

xyzpold [p,3] (double) Set by user.

Old values of xyzp. Set with c: copy xyzp xyzpold after the control volumes are reset but before xyzp is reset.
Needed by c_ppreset. (12/14)

zrotation [1 or 3] (double) Set by the user.
zrotation[0] is the rotation rate about the z axis.. If *zrotation* is dimensioned 3, the axis of rotation is at x=*zrotation*[1], y=*zrotation*[2].

If available, *zrotation*[0] is used by c_bijrhsmarv(s) ex_ bijrhsmarvex, c_ddtall, c_momcamddt, c_momrhsr. If *zrotation* is dimensioned 3, c_momrhsr includes the centrifugal term (as well as the Coriolis which depends only on *zrotation*[0]). (12/14)