

## Class Descriptions for Mix2dsp

### Class CInputs (Input parameters)

Member functions:

- GetVar(parameter) – Returns the type-double value for the specified parameter
- SetVar(parameter, value) – Sets the specified parameter to the type-double value

List of variables for CInputs:

Variable [Units]	Variable Name
D effective [pm/V]	DEFF
Delta k [1/mm]	DELTAK
Energy [J] (s,i,p)	ENERGY_S, ENERGY_I, ENERGY_P
Beam diameters [mm] (s,i,p)	BEAMDIAM_S, BEAMDIAM_I, BEAMDIAM_P
Pulse duration [ps] (s,i,p)	DURATION_S, DURATION_I, DURATION_P
Wavelengths [nm] (s,i,p)	LAMBDA_S, LAMBDA_I, LAMBDA_P
Index of refraction (s,i,p)	INDEX_S, INDEX_I, INDEX_P
Group velocity index (s,i,p)	GVI_S, GVI_I, GVI_P
Group velocity dispersion (s,i,p)	GVD_S, GVD_I, GVD_P
Crystal absorption [1/mm] (s,i,p)	ALPHA_S, ALPHA_I, ALPHA_P
Crystal input reflectivity (s,i,p)	RC_IN_S, RC_IN_I, RC_IN_P
Crystal output reflectivity (s,i,p)	RC_OUT_S, RC_OUT_I, RC_OUT_P
Crystal length [mm]	CRYSTLENGTH
N2 signal [sq cm/W] (s,i,p)	N2_S1, N2_S2, N2_S3
N2 idler [sq cm/W] (s,i,p)	N2_I1, N2_I2, N2_I3
N2 pump [sq cm/W] (s,i,p)	N2_P1, N2_P2, N2_P3
Beta signal [cm/W] (s,i,p)	BETA_S1, BETA_S2, BETA_S3
Beta idler [cm/W] (s,i,p)	BETA_I1, BETA_I2, BETA_I3
Beta pump [cm/W] (s,i,p)	BETA_P1, BETA_P2, BETA_P3
Walkoff angles [mrad] (s,i,p)	RHO_S, RHO_I, RHO_P
Radius of curvature [mm in air] (s,i,p)	RADCURV_S, RADCURV_I, RADCURV_P
Offset in walkoff direction [mm] (s,i)	OFFSET_S, OFFSET_I
Number of grid points/zsteps (x,y,z,t)	NX, NY, NT, NZ
Maximum grid extent [mm] (x,y)	MAXGRID_X, MAXGRID_Y
Gaussian index order	GAUSSINDEX_S, GAUSSINDEX_I, GAUSSINDEX_P
Phases [rad] (s,i,p)	PHI_S, PHI_I, PHI_P
Time delay relative to pump [ps]	DELAY_S, DELAY_I
Distance to detector [mm]	DIST

### **Class CGlobVar (Run-time calculated parameters)**

Member functions:

- dSetGlobal(parameter, value) – Sets the parameter to a double value
- ulSetGlobal(parameter, value) – Sets the parameter to an unsigned long value
- dGetGlobal(parameter) – Returns a double value for the specified parameter
- lGetGlobal(parameter) – Returns an unsigned long value for the specified parameter

List of variables for CGlobalVar:

Variable [Units]	Variable Name
Frequencies [rad/s] (s,i,p)	OMEGA_S, OMEGA_I, OMEGA_P
Step sizes (x,y,z,t)	DX, DY, DZ, DT
Intensity term ( $-4 \cdot \log(0.5)/\pi$ )	I_TERM
Input peak power [W] (s,i,p)	PEAKPOW_S, PEAKPOW_I, PEAKPOW_P
Input peak irradiance [J/sq m] (s,i,p)	PEAKIR_S, PEAKIR_I, PEAKIR_P
Input peak fluence [W/sq m] (s,i,p)	PEAKFL_S, PEAKFL_I, PEAKFL_P
Output peak irradiance [J/sq m] (s,i,p)	IROUT_S, IROUT_I, IROUT_P
Output pulse energy [J] (s,i,p)	EPULSE_S, EPULSE_I, EPULSE_P
Grid size vector (x,y,t) unsigned long	NN
Recalculated grid sizes for the graph (x,y)	NGRIDX, NGRIDY
Pad value for the fluence graph	PADVAL
Delta theta for farfield graph (x,y)	DTHETAX, DTHETAY
New calculated origin for fluence graph (x,y)	ORIGINX_FL, ORIGINY_FL

### **Class CFields (Electric fields)**

Class constructor:

CFields Name\_field(gridtotal(nx\*ny\*nt), nx, ny, nt)

Member functions:

- fSetField(field name, array\_element, value) – Sets the specified array\_element for a type-float field to a type-float value
- dSetField(field name, array\_element, value) – Sets the specified array\_element for a type-double field to a type-double value
- fGetField(field name, array\_element) – Returns the type-float value for the field's specified array\_element
- dGetField(field name, array\_element) – Returns the type-double value for the field's specified array\_element

List of Variables for CFields:

Variable	Variable Name
Electric fields [V/m]	EFIELD_S, EFIELD_I, EFIELD_P
Phases	PHASE_S, PHASE_I, PHASE_P
Phase term for the exponential	PHASETERM_S, PHASETERM_I, PHASETERM_P
Grids (x,y,t)	XGRID, YGRID, TGRID
Supergaussian coefficients	SGCOEFF
Arrays for the FFT's	FFTW, SPECFFT, FARFFT_X, FARFFT_Y

## **Class CBeams (Beam analysis)**

Class constructor:

CBeams Name(number of time points)

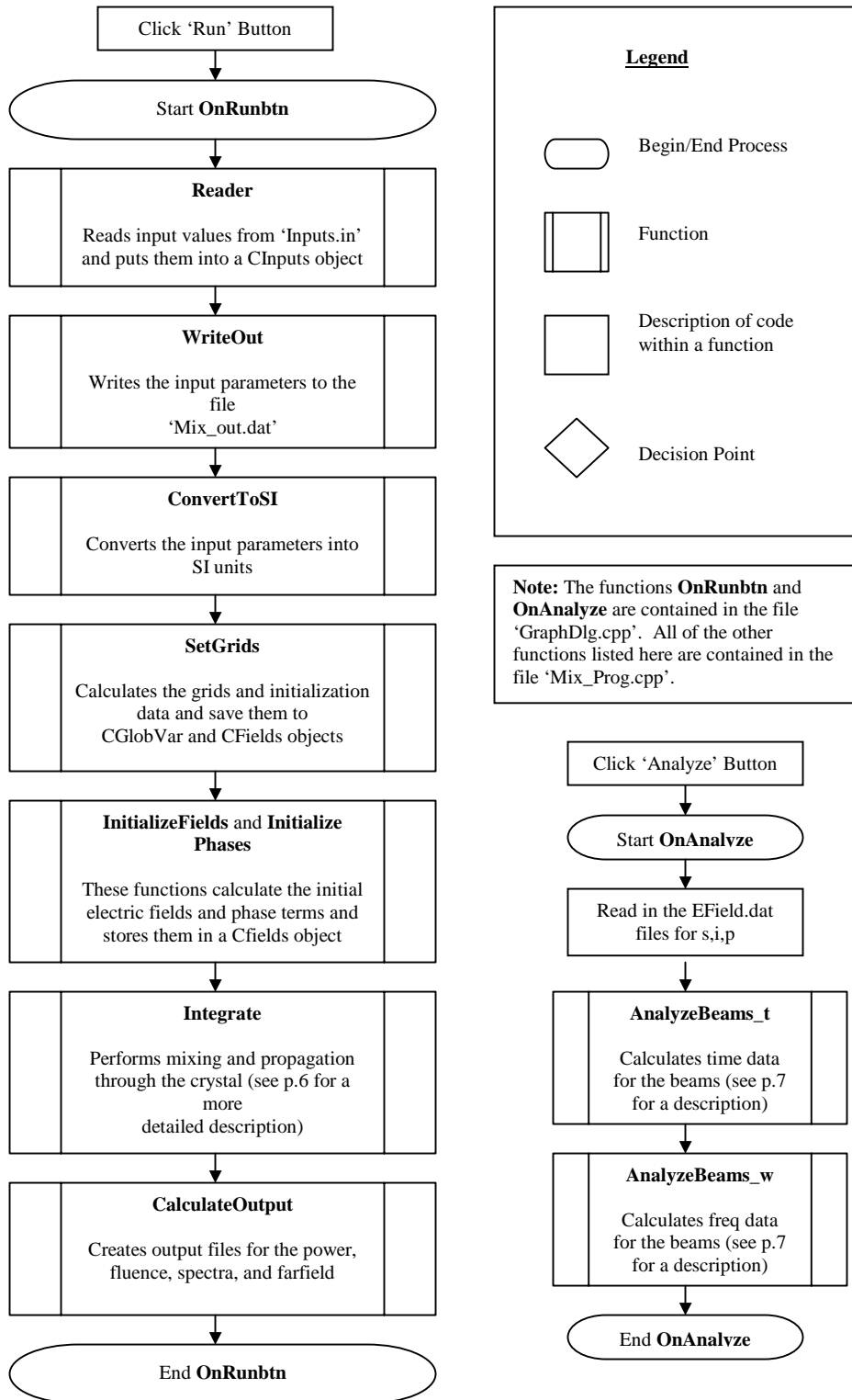
Member functions:

- SetBeamx(array\_element, tilt, msqr, zo, wavofcurv, sigsqx, sigsqs, asubx, xbar) – Sets the array\_element to the beam values in the walkoff direction
- SetBeamy(array\_element, tilt, msqr, zo, wavofcurv, sigsqx, sigsqs, asubx, xbar) – Sets the array\_element to the beam values perpendicular to the walkoff direction
- GetTiltx(array);
- GetTilty(array);
- GetMsqr(x,array);
- GetMsqy(y,array);
- GetZOx(array);
- GetZOy(array);
- GetWavofCurvx(array);
- GetWavofCurvy(array);
- GetSigsqx(array);
- GetSigsqy(array);
- GetSigsqs\_x(array);
- GetSigsqs\_y(array);
- GetAsubx(array);
- GetAsuby(array);
- GetXbar(array);
- GetYbar(array);

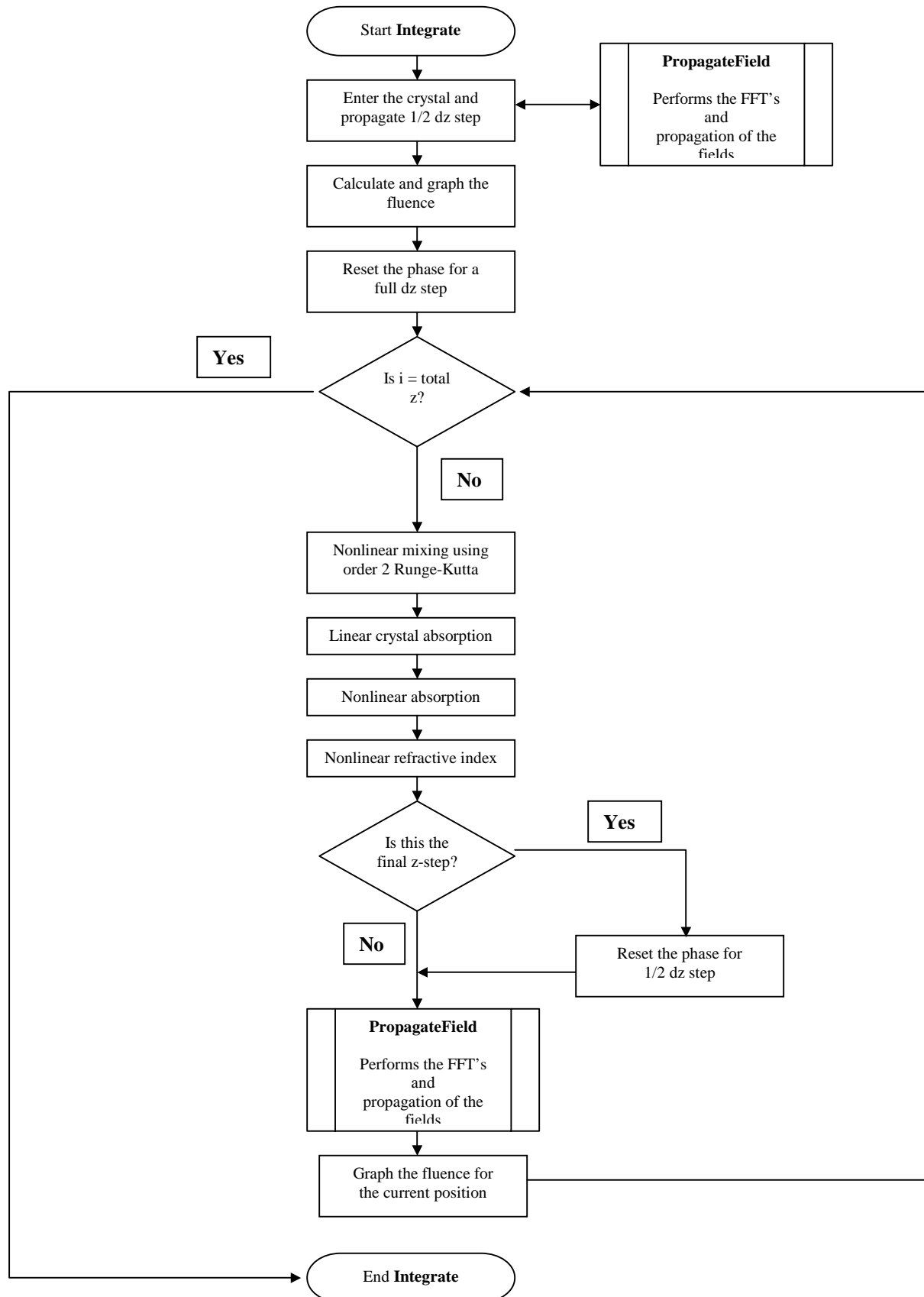
List of variables for CBeams:

Variable [Units]	Variable Name
Beam tilt in x and y	pTiltx, pTilty
M-squared in x and y	pMsqr(x), pMsqr(y)
Distance to waist in x and y	pzo(x), pzo(y)
Waves of curvature in x and y	pWav(x), pWav(y)
Sigma squared in x and y	pSigsqx, pSigsqy
Sigma squared s in x and y	pSigsqs_x, pSigsqs_y
Asubx and Asuby	pAsubx, pAsuby
Xbar and Ybar	pXbar, pYbar

## Flowchart for the Mix2dsp Program



## Flowchart for Function Integrate



## Flowchart for Functions AnalyzeBeams\_t and AnalyzeBeams\_w

