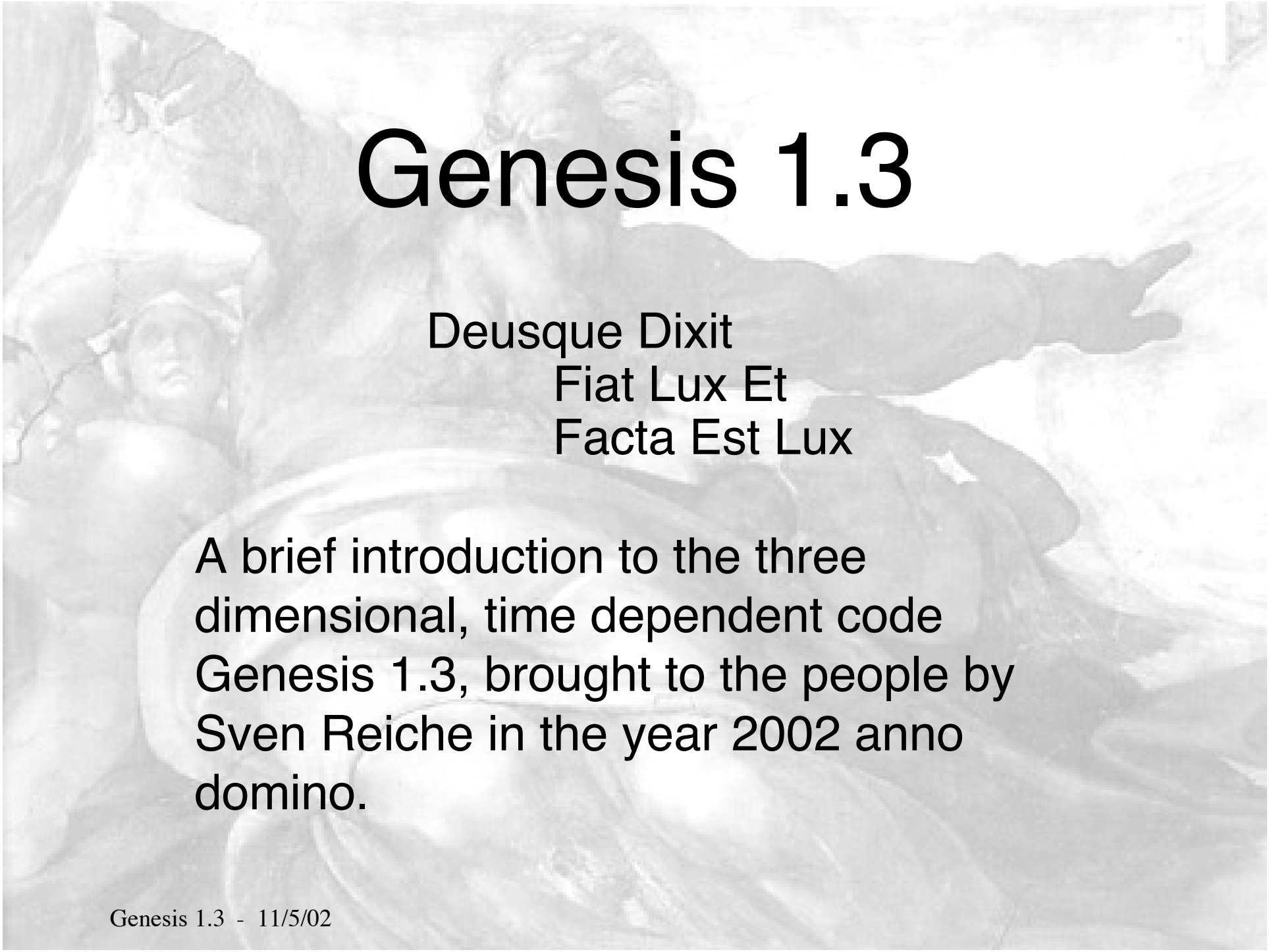


Genesis 1.3

Deusque Dixit
Fiat Lux Et
Facta Est Lux

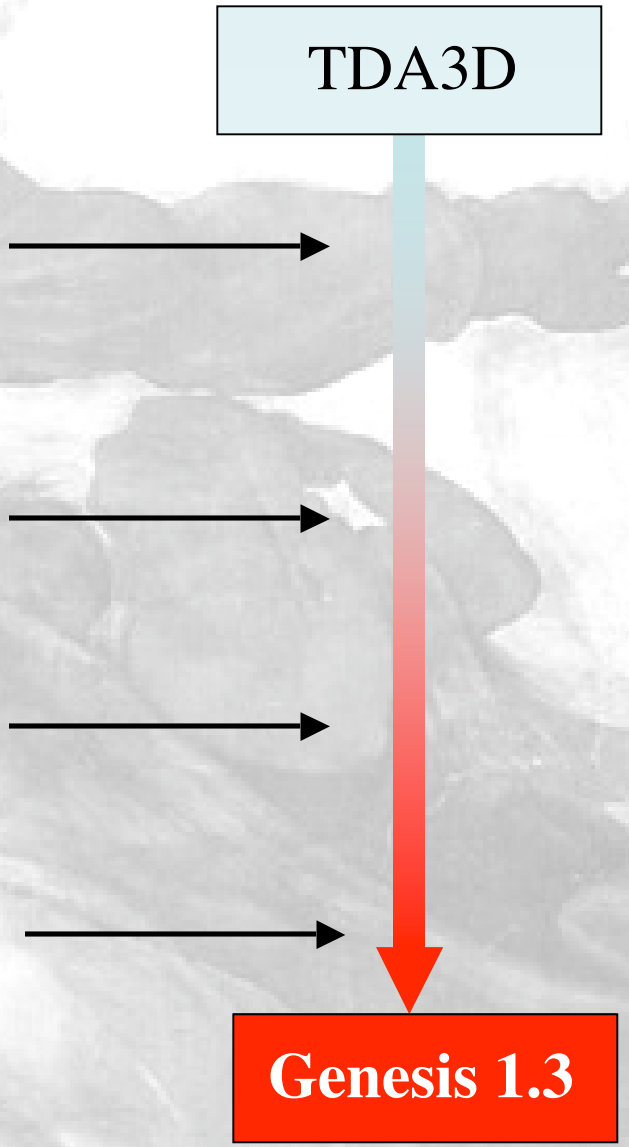
A brief introduction to the three dimensional, time dependent code Genesis 1.3, brought to the people by Sven Reiche in the year 2002 anno domino.

Overview

- History
- Features
- Part I - Numerics
  - Field Solver
  - Particle Solver
  - Phase Space Loading
  - Time-dependent Simulations
- Part II - Running Genesis 1.3
  - Input Parameters
  - Input/Output Files
  - Running Genesis 1.3

History

Need for 3D time-dependent code



Replace (r,□) grid with Cartesian grid
New field solver

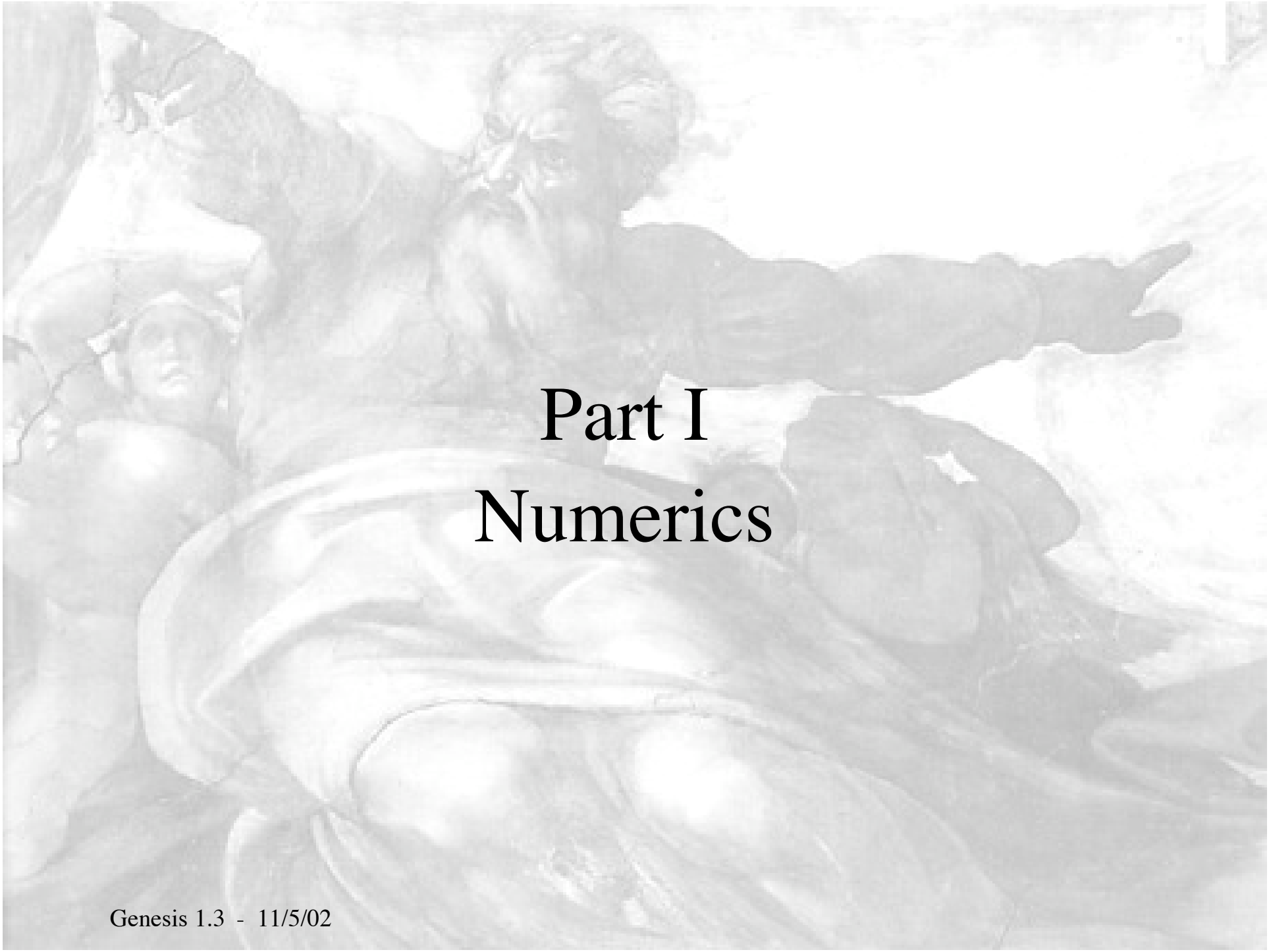
New particle solver for transverse coordinate, optimization for longi-tudinal variables

Time-dependence + shot noise algorithm

External magnet, radiation & electron beam files

Features

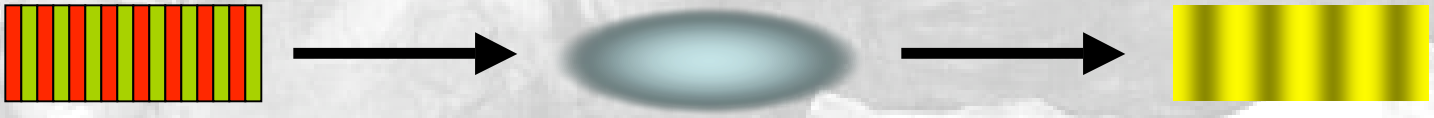
- Solves eikonal field equation (slow varying amplitude).
- Field discretized on fully Cartesian grid (steady-state simulation requires only transverse grid).
- Fully 6 dimensional tracking of electron beam (equations of motion averaged over undulator period).
- Runs in steady-state, time-dependent and scan mode.
- External input of magnetic lattice, electron distribution and seeding radiation pulse



Part I
Numerics

FEL Simulations

The extreme scales in FEL make standard EM-codes impractical.



Undulator
(1-100 m)

Beam
(0.1-1 mm)

Wavelength
(1 Å - 10 μm)

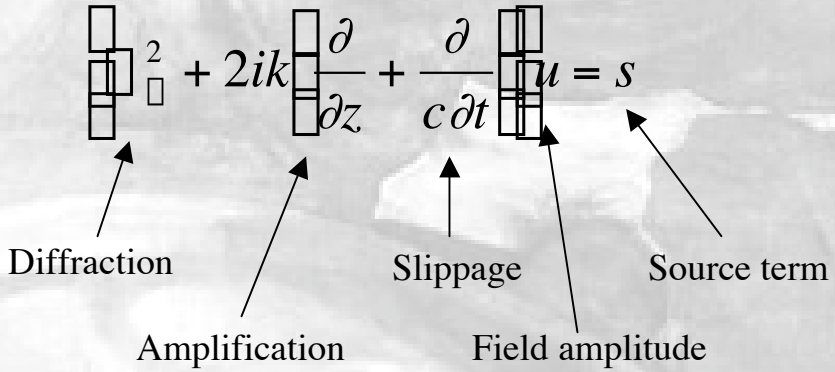
LCLS covers 12 orders of magnitudes

Solutions/Approximation

- Co-moving frame (no global grid)
- Par-Axial approximation (no fast oscillating terms)
- Period-averaged equations of motion (larger step sizes)

Field Equation

Differential equation for radiation field in eikonal approximation is:



Field equation very similar to Schroedinger equation

Various methods exist to solve it:

- Fast
- Memory efficient
- Discretization (no continuous solutions)

Field Discretization

- Cartesian grid in x and y with grid spacing Δ
- Integration step Δz along undulator
- Slippage and discretization in t ignored for now
- Numbered grid points ($k(i,j)$)

$$u(x,y,z) \rightarrow \tilde{u}^m_{i,j} \equiv u^m_k = \mathbf{u}^m$$

Discretized differential operators

$$\frac{\partial}{\partial z} u \rightarrow \frac{u^{m+1}_{i,j} - u^m_{i,j}}{\Delta z}$$

$$\nabla^2_{\perp} u \rightarrow \frac{u^m_{i+1,j} + u^m_{i-1,j} + u^m_{i,j+1} + u^m_{i,j-1} - 4u^m_{i,j}}{\Delta^2}$$

Laplace operator is a Matrix, operating on vector **u** at step m

Discretized Field Equation

Laplace operator applied to equally weighted field of step *m* and *m*+1. (Crank-Nicholson Scheme). More implicit schemes, where the field at *m*+1 is weighted stronger, are more stable but less precise, while explicit schemes with stronger weight of the field at *m* are instable.

$$\left(\mathbf{I} - i\frac{\Delta z}{4k}\mathbf{L}\right)\mathbf{u}^{m+1} = \left(\mathbf{I} + i\frac{\Delta z}{4k}\mathbf{L}\right)\mathbf{u}^{m} + s^{m+1/2}\Delta z$$

Problem is reduced to inverting matrix $\mathbf{K}=\mathbf{I}-(i\Delta z/4k)\mathbf{L}$. Note that $\mathbf{K}$ is sparse, but the inverse $\mathbf{K}^{-1}$ is not. For 10^4 grid points, $\mathbf{K}$ has $4\cdot10^4$ non-zero elements, $\mathbf{K}^{-1}$ has 10^8 (memory problem).

- Alternating Direction Implicit Method
- Relaxation Method

ADI Integration

Laplace operator **L** is divided into two sub-matrices $\mathbf{L}_x$ and $\mathbf{L}_y$, where each sub-matrix acts only in the direction, indicated by the index. The integration is divided into the same number of sub-steps. When for all steps only one implicit integration occurs for each direction, the integration unconditionally stable.

$$\left(\mathbf{I} - i\frac{\Delta z}{4k}\mathbf{L}_x\right)\mathbf{u}^{m+1/2} = \left(\mathbf{I} + i\frac{\Delta z}{4k}\mathbf{L}_y\right)\mathbf{u}^{m} + s^{m+1/2}\frac{\Delta z}{2}$$

$$\left(\mathbf{I} - i\frac{\Delta z}{4k}\mathbf{L}_y\right)\mathbf{u}^{m+1} = \left(\mathbf{I} + i\frac{\Delta z}{4k}\mathbf{L}_x\right)\mathbf{u}^{m+1/2} + s^{m+1/2}\frac{\Delta z}{2}$$

The matrices $\mathbf{K}_x=\mathbf{I}-(i\Delta z/4k)\mathbf{L}_x$ and $\mathbf{K}_y=\mathbf{I}-(i\Delta z/4k)\mathbf{L}_y$ are tri-diagonal and allow for a memory efficient and fast calculation.

Space Charge

Only longitudinal space charge is considered

- Decomposition into Fourier series in ζ (longitudinal position) and θ (azimuthal angle)
- 1D grid centered to beam centroid for each integration step

$$\left[\mathbf{L}_m - l^2 \frac{k^2(1+a_u^2)}{\gamma_R^2}\mathbf{I}\right]\mathbf{E}_{l,m} = \rho_{l,m}$$

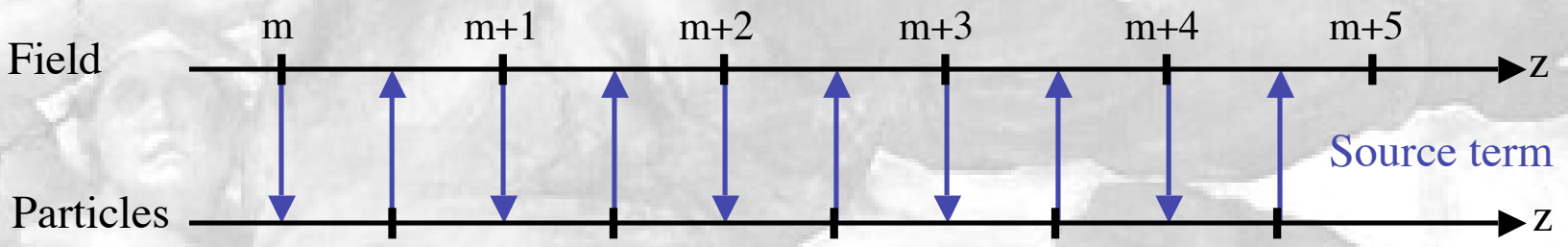
Although different from the Cartesian, discretized Laplace operator, $\mathbf{L}_m$ is tri-diagonal.

+ Fast solver (~100 grid points, a few longitudinal & azimuthal modes).

- Slow construction of source term $\rho_{l,m}$ due to numerous calls to harmonic functions and their inverse

Particle Solver

Leap frog integration of field and particle. Best precision, when source term is evaluated in the middle of the integration step.



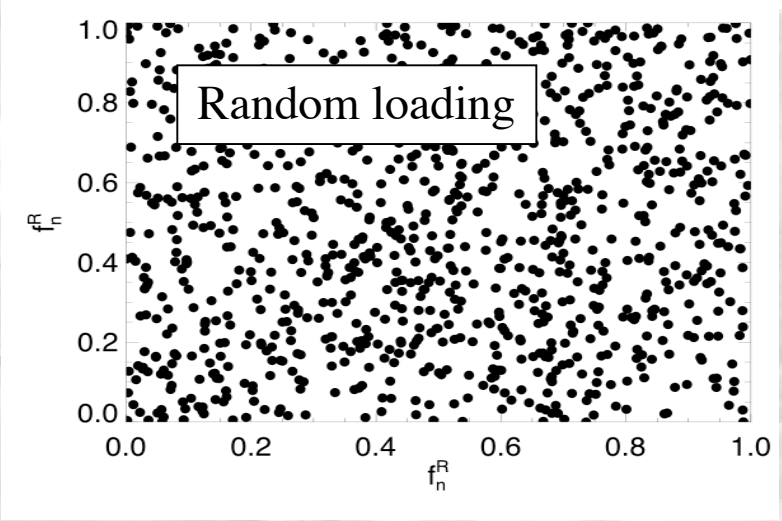
Energy & Particle phase (longitudinal position) are solved with 4th order Runge-Kutta solver. Best method for small, fixed integration step sizes. The leap frog system prohibits adaptive schemes.

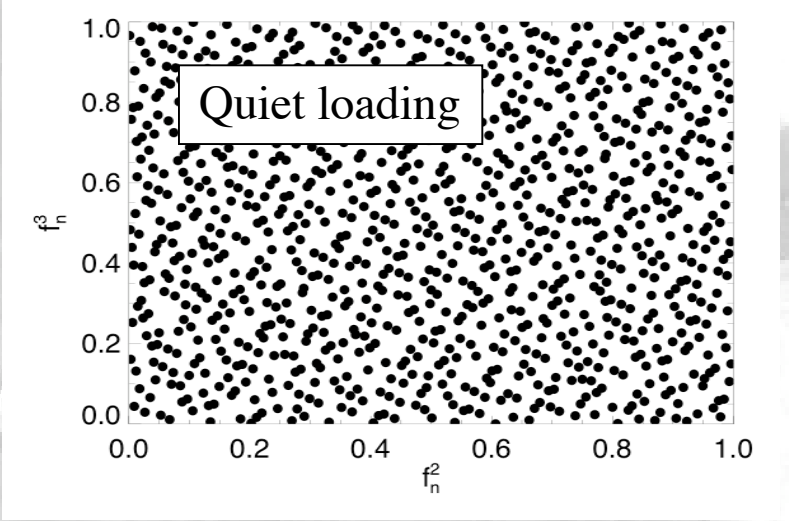
Differential equations are average over undulator period to allow longer integration step sizes.

Transverse variables are advanced by matrix multiplication up to first order.

Phase Space Loading

Smooth filling phase space with Hammersley sequences:





Algorithm (n should be prime number to avoid correlation)

Count	n-base number (n=2)	Invert digit	Convert back
1	1	0.1	0.5
2	10	0.01	0.25
3	11	0.11	0.75
4	100	0.001	0.125
…	…	…	…

Phase Space Loading II

1. Fill all 6 dimension using Hammersley sequences.
2. Transform to different distribution (Gaussian, parabolic, uniform), using methods of inversion function, rejection or joint probability.
3. Mirror distribution in x, y, p_x and p_y to exclude moments in transverse distribution.
4. Shift, scale and correlate (e.g. β_x and β_y) transverse variables and energy for final distribution.
5. Duplicate entire distribution with offsets in phase (bins) to eliminate residual moments in bunching.
6. Apply shot noise

Alternative, import external distribution (Step 5 & 6 still required)

Shot Noise

Shot noise algorithm has to enforce the correct statistic!

Local bunching factor b_j of the jth slice is proportional to the emission of the spontaneous radiation and must depend on ***n*** - the number of electrons to be simulated - and not on ***N*** - the number of macro particle used.

$$b_j = \frac{1}{N} \sum_{k=1+jN}^{(j+1)N} e^{i\phi_k}$$

Required statistic of bunching factor (n >> N):

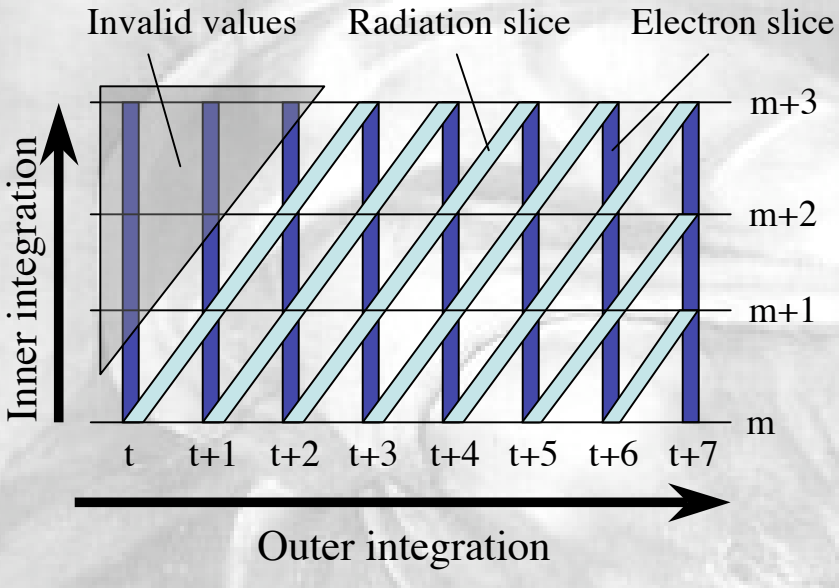
$$\langle b_j \rangle = 0 \qquad \langle |b_j|^2 \rangle = \frac{1}{n} \qquad |b_j|^2 \text{ follows neg. distribution}$$

Penman-Algorithm gives correct statistic by adding random phase offset between 0 and √(3n/N) to each macro particle. Works fine for the fundamental.

Time-dependent Simulations I

Extending the grid in the *t*-direction would require the entire field to be held in memory. Approximation by a fully explicit integration in *t*, suppressing propagation in the backward direction.

$$\left(\frac{\partial}{\partial z}+\frac{\partial}{c\,\partial t}\right)u \rightarrow \frac{\mathbf{u}^{m+1,t+1}-\mathbf{u}^{m,t}}{\Delta z}$$



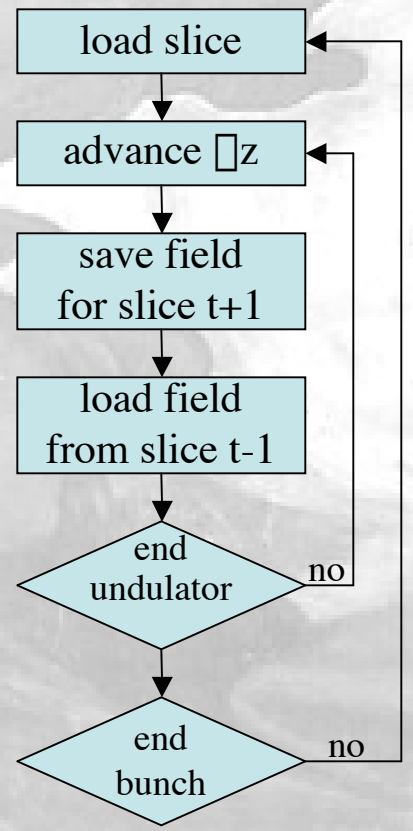
Starting simulation with slice t=1, emission from slices t<1 unknown. Simulation becomes valid after a radiation slice has interact with electron slice at every integration step!

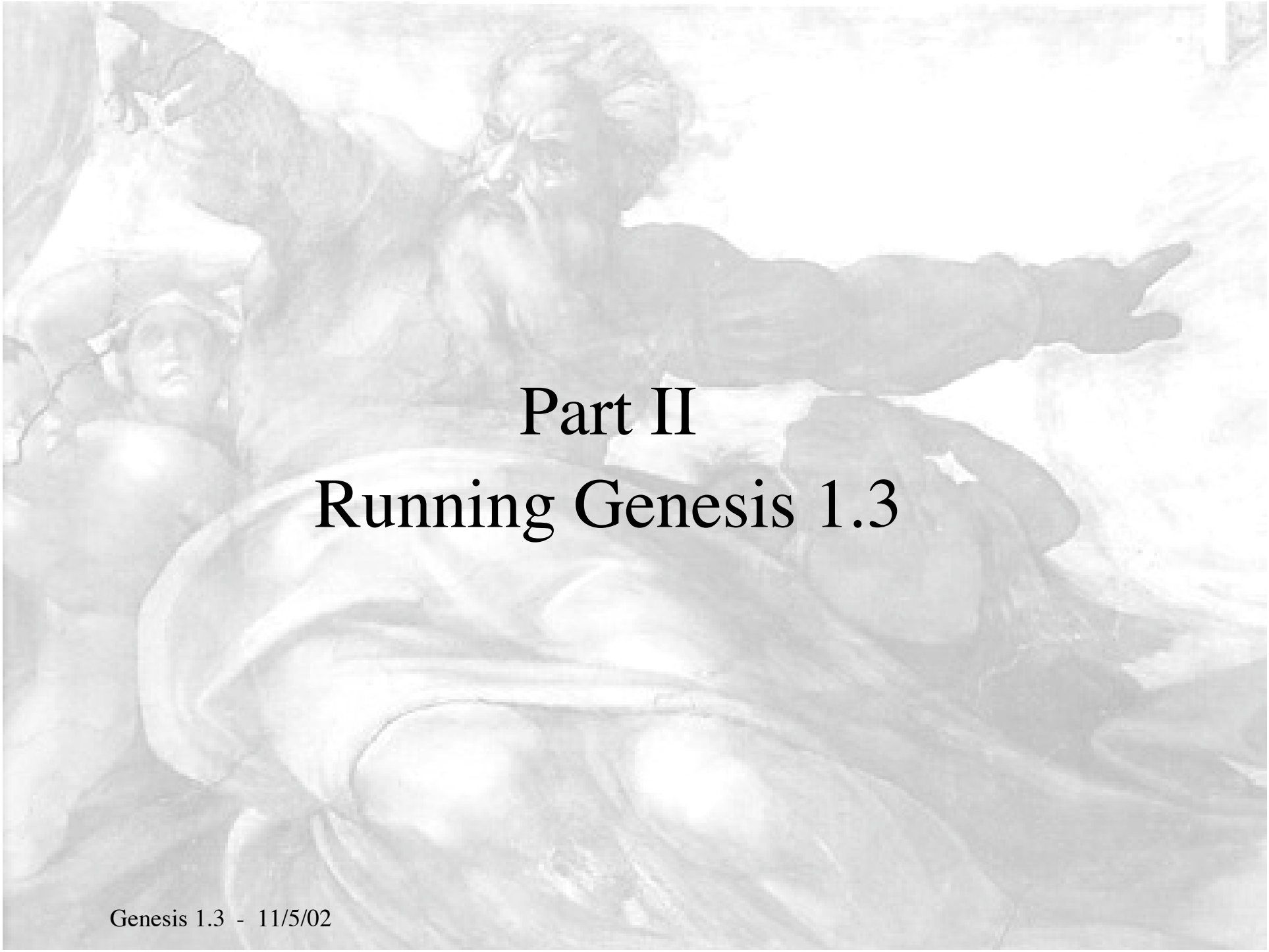
Time-dependent Simulations II

Simulation not limited by number of slices.

- Enough slices to cover unphysical region
- Even more slices if spectrum is evaluated
- Step size must be much smaller than gain length to keep approximation for time-dependent simulation valid!
- # macro particles per slice is fixed and is not a measure for local current:
  - Difficult to import external distribution.
  - Better efficiency for parallel version of Genesis 1.3

Algorithm





Part II
Running Genesis 1.3

The Genesis 1.3 Program

- Single executable file.
- Runs in a shell environment.
- Base version requires no additional system libraries.
- Can be built from FORTRAN source code on any platform.
- Current version is 1.0, soon to be released (hopefully before Christmas)
- Source code + Linux executable can be downloaded from website http://pbpl.physics.ucla.edu/genesis
- Be prepared! - Genesis 1.3 can easily produce hundreds of MByte of data.

Minimal Run

```
reiche@colt45:~/genesis_runs$ genesis
------------------------------
GENESIS 1.3 HAS BEGUN EXECUTION
(Version 1.0 Unix)

PLEASE ENTER INPUT FILE NAME
visa.in
PLEASE ENTER OUTPUT FILE NAME
visa.out
Slice     1: Simulation  10% completed.
Slice     1: Simulation  20% completed.
Slice     1: Simulation  30% completed.
Slice     1: Simulation  40% completed.
Slice     1: Simulation  50% completed.
Slice     1: Simulation  60% completed.
Slice     1: Simulation  70% completed.
Slice     1: Simulation  80% completed.
Slice     1: Simulation  90% completed.
Slice     1: Simulation 100% completed.
***  Writing history record for slice     1
***  Closing files

GENESIS RUN HAS FINISHED
------------------------
reiche@colt45:~/genesis_runs$
```

Start Genesis 1.3

Prompt for main input file

Name of output file
(overwrites existing files!)

Progress report

Confirms that output is written

Main Input File

- FORTRAN namelist, starting with the tag `$newrun` and ends with `$end`.
- Parameters, not defined in the namelist, fall back on their default value.
- Any text outside the namelist tags is ignored
- If input file is not found, Genesis 1.3 create the template `template.in`, containing all possible input parameters.

```
$newrun
 AW0    =  8.910000E-01
 XLAMD  =  1.800000E-02
 NPART  =  8192
 GAMMA0=   1.415000E+02
 XLAMDS=   8.083000E-07
 NCAR   =  141
 NWIG   =  224
 DELZ   =  1.000000E+00
 CURPEAK=   1.500000E+02
 $end
```

- Parameter are case insensitive
- Parameters in a single line are separated by commas
- First character has to be space

Input Parameters

Genesis 1.3 accepts about 100 input parameters. They are loosely grouped by functionality into

- Undulator parameters
- Electron beam parameters
- Radiation parameters
- Focusing parameters
- Time-dependence parameters
- Control parameters
- Grid parameters
- IO parameters

Some parameters have secondary meanings (e.g. `ZRAYL`)

Setting-up Undulator

`AW0`	RMS undulator parameter
`XLAMD`	Undulator period
`NWIG`	Periods of undulator module
`NSEC`	# undulator modules
`XKX, XKY`	Relative natural focal strength in *x* and *y*
`IWITYP`	0 = planar, 1 = helical
`AWD`	Effective undulator parameter for gaps

The sum of `XKX` and `XKY` has to be 1 for any real undulator, although GENESIS 1.3 accepts any value.
Fine matching of modules is done with `AWD`, because the integration step size cannot be varied.

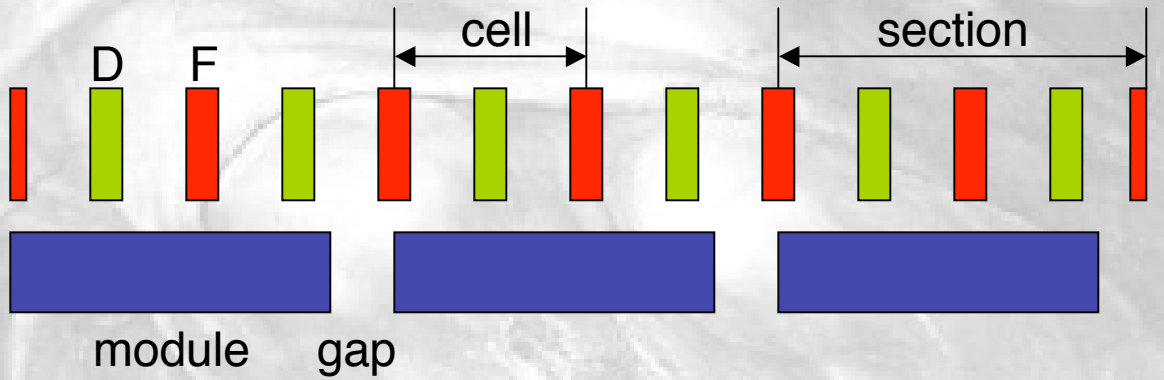
Setting-up Magnet Lattice

`QUADF,QUADD`	Quadrupole field strength in T/m
`FL,DL,DRL`	Quad & drift lengths in measure of `XLAMD`
`F1ST`	Start position of FODO cell with respect to beginning of F-quadrupole

Genesis 1.3 aligns undulator modules to the beginning of a FODO cell, causing drift section between modules.

Example:

`FL`=`DL`=2, `DRL`=4, `F1ST`=1, `NWIG`=20, `NSEC`=3



Setting-up Electron Beam

`NPART`	#macro particle per slice
`GAMMA0`	Mean energy
`DELGAM`	RMS energy spread
`RXBEAM, RYBEAM`	Beam size in x and y
`EMITX, EMITY`	Normalized emittance
`ALPHAX, ALPHAY`	Twiss parameter
`CURPEAK`	Current in Amperes

All parameters, except for `CURPEAK,` are the same for all slices in time-dependent simulations.

Setting-up Radiation Field

XLAMDS	Radiation wavelength (no necessarily resonant)
PRAD0	Input radiation power in Watts
ZRAYL	Rayleigh length of seeding radiation field
ZWAIST	Waist position of seedin radiation field.

ZRAYL and ZWAIST are only used to calculate the initial size of the seeding, fundamental Gaussian mode with

$$w_0 = \sqrt{\frac{\mathrm{XLAMDS} \cdot \mathrm{ZRAYL}}{\pi}\left(1 + \left(\frac{\mathrm{ZWAIST}}{\mathrm{ZRAYL}}\right)^2\right)}$$

Because w_0 defines the grid size, ZRAYL and ZWAIST are required even for SASE simulations.

Setting-up Grid

NCAR	Grid points in one direction (must be odd)
RMAX0	Relative size of grid
DGRID	Absolute size of grid (overwrites RMAX0)
NSCZ	Fourier coefficient in z for space charge (0=disabled)
NSCR	Fourier coefficient in f for space charge
NPTR	Grid points for space charge field

Genesis 1.3 calculates the grid size [-□, □] by

$$\square = \texttt{RMAX0} \cdot \left(\texttt{RXBEAM} + \texttt{RYBEAM} + w_0 \right)$$

or □=DGRID, when DGRID is defined in the input deck. SASE simulations should run with a large grid thanstead-state simulations.

Setting-up Control Parameters

DELZ	Step size in units of XLAMD
ZSTOP	End of integration length (0 = undulator length)
NBINS	Bins in longitudinal coordinate □
IALL	Enforces same seed of Hammersley sequences for all beam slices
IPSEED	Seed for shot noise random number generator
SHOTNOISE	0 = disable shot noise

DELZ must resolve all magnets. Sometime it requires that magnet length hast to be adjusted, while keeping the product QUADF·FL constant.

Setting-up Time-dependence

`ITDP`	Non-zero value enables time-dependence
`CURLEN`	positive: RMS length of Gaussian profile negative: FWHM length of step profile
`ZSEP`	Thickness of slices in units of `XLAMDS`
`NSLICE`	Total number of slices
`NTAIL`	Position of 1st slice with respect to t=0
`IOTAIL`	Non-zero value enables output of 1st slippage length

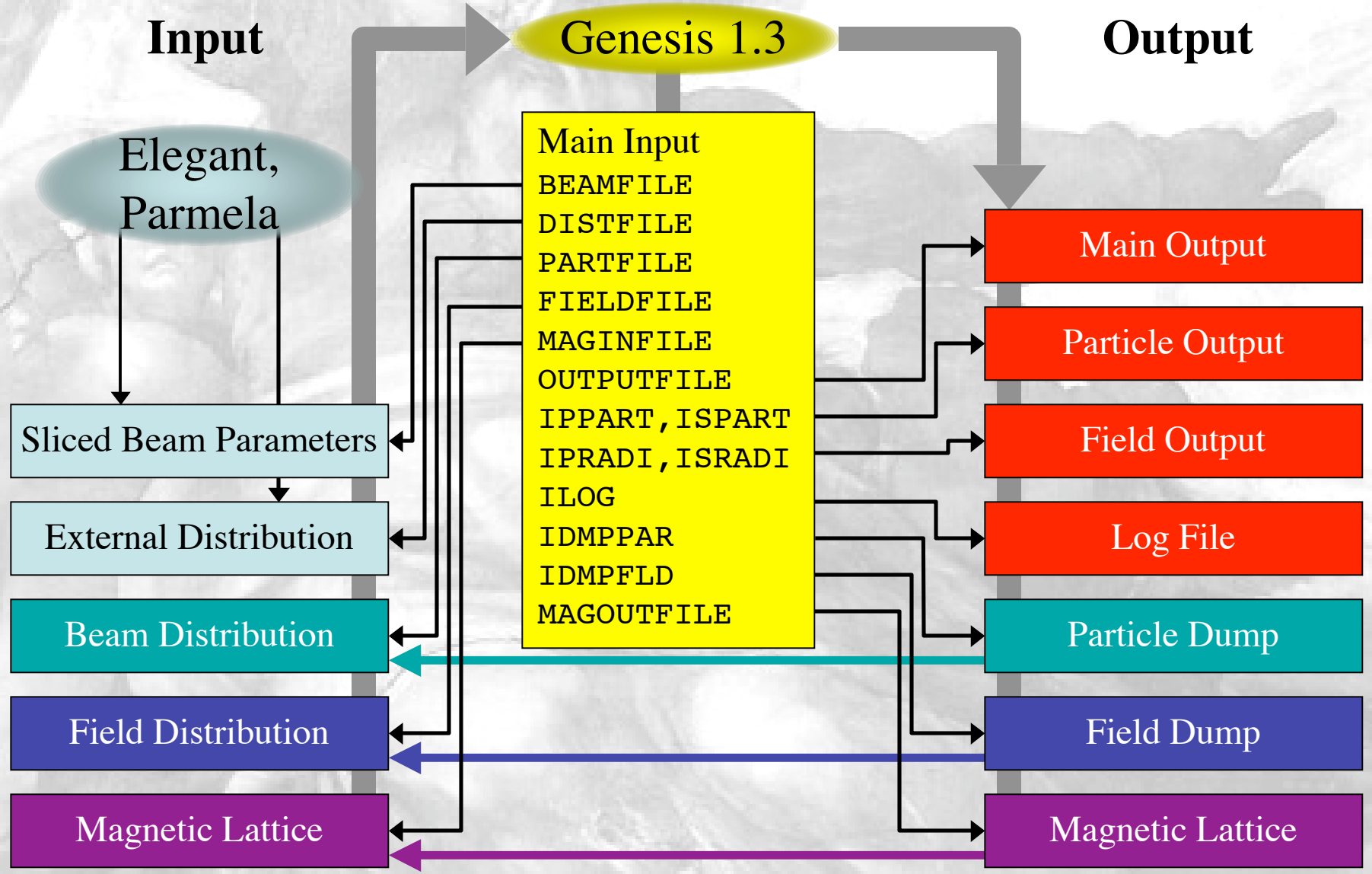
To simulate the time-window $[t_0,t_1]$:

`NSLICE`= $c(t_1\text{-}t_0)$/`ZSEP`/`XLAMDS`

`NTAIL`= ct_0/`ZSEP`/`XLAMDS`

A step profile starts at t = 0s while the Gaussian profile is centered at that point.

Input/Output Files



Magnetic Lattice Files

ASCII file, listing components in sequential order. Order of different components can be mixed. If a component is not defined in the file, it is generated internally by GENESIS 1.3. Field error and taper are always enforced.

```
# LCLS lattice
? VERSION = 1.0
#
? UNITLENGTH = 3.E-2
!LOOP=50
AW    2.623    64.   8.
!ENDLOOP
QF    2.2750E+01   8. 0.
!LOOP=25
QF   -2.2750E+01   8. 64.
QF    2.2750E+01   8. 64.
!ENDLOOP
```

Comments start with #

`VERSION = 1` allows more flexible input

Base length, everything is scaled to

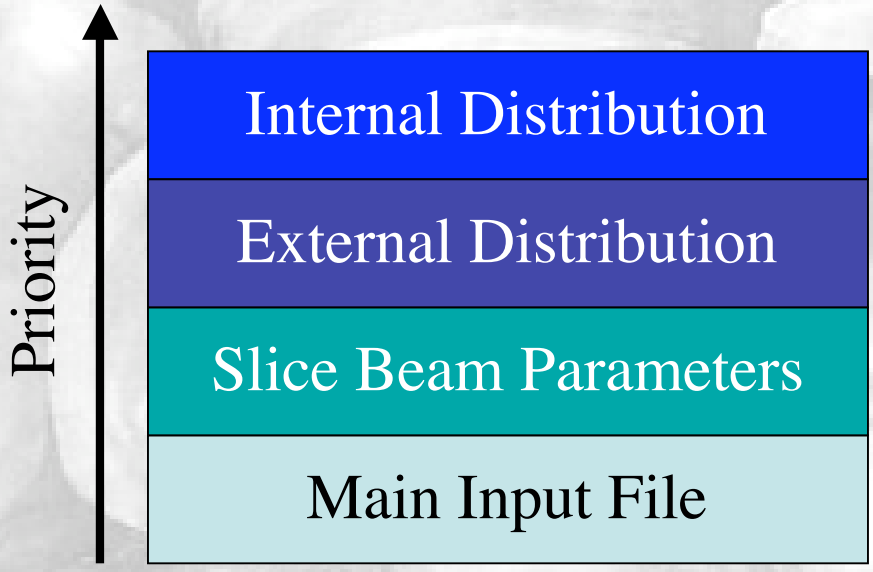
Unrolls 50 times everything between `!LOOP` and `!ENDLOOP`

Data line, must contain always 4 elements: ID, strength, length & offset to previous element

(AW = undulator, QF = quadrupole)

Electron Beam Input I

Genesis 1.3 accepts different levels of electron beam input. Input with higher priority (partially) overwrites input of lower priorities. Internal distribution are bets generated by Genesis 1.3 itself (particle dump) and then modified.



Directly imports 6D phase space

Fills up 5D phase space + add phase □

Local beam parameters + wakes

Global beam parameters + local current

Electron Beam Input II

Sliced Beam Parameters (interpolation from look-up table)

```
? VERSION = 1.0
? SIZE = 201
? COLUMNS ZPOS GAMMA0 CURPEAK
  -0.001500  140.792 74.6741
  -0.001485  140.800 75.7179
  -0.001470  140.807 76.7655
…
```

Flexible input

Size of table (optional)

Columns of table

Table

`elegant2genesis` *creates table from Elegant distribution*

External Distribution

```
? VERSION = 1.0
? CHARGE = 170e-12
? SIZE =  63291
? COLUMNS X PX Y PY T P
 5.244632846321E-004 -8.470245142763E-002 …
 5.086302515453E-004 -6.127686199672E-002 …
…
```

Required

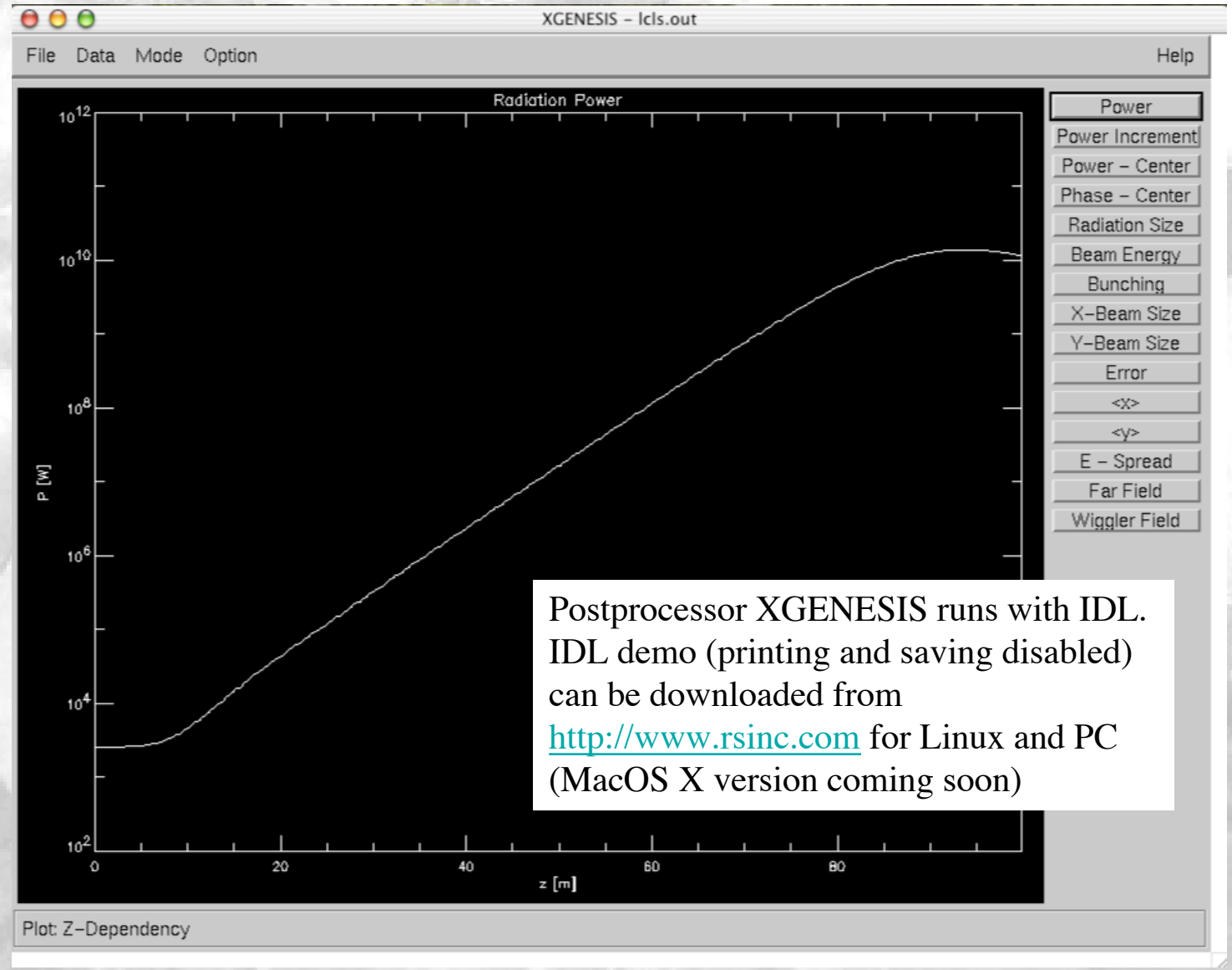
Other columns are `XPRIME` instead of `PX`, `Z` instead of `T`, `GAMMA` instead of `T`. Any other label causes that the column is ignored.

Output

Main output file is split into 4 parts:

1. Copy of main input file
2. Data to describe size of following output data as well as particle and field output (binary format)
3. Global data (position in undulator, magnet lattice)
4. Sequential list of all slices
    1. Slice number
    2. Local current
    3. Table with main output (e.g. radiation power, phase, increment size, electron beam energy, spread, size, bunching)

Postprocessor



Postprocessor XGENESIS runs with IDL. IDL demo (printing and saving disabled) can be downloaded from http://www.rsinc.com for Linux and PC (MacOS X version coming soon)

Outlook

- Release of version 1.0
- Support of SDDS, XML and HDF for interface with other programs and easier scripting.
- Parallel version of Genesis 1.3, using MPI
- New postprocessor for Genesis 1.3
- Sample files for LCLS and VISA at the course web site.