VUV and X-ray Free-Electron Lasers

Time-independent 3D FEL Simulations

Petr Anisimov,[2] Nicole Neveu[1], Dinh C. Nguyen[1]

[1] SLAC National Accelerator Laboratory

[2] Los Alamos National Laboratory

LA-UR-21-20607

Thursday (Jan 28) Lecture Outline

- Transverse dynamics and natural focusing
- Q&A
- Strong focusing and a FODO lattice
- Q&A
- Ming Xie analysis
- Q&A
- Numerical simulator – LUME-Genesis
- Q&A

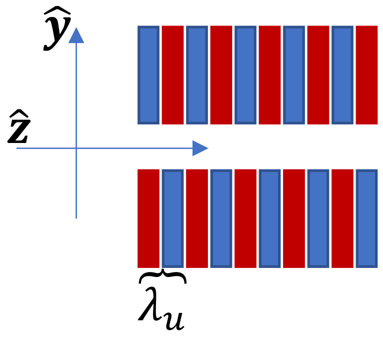
Transverse dynamics and natural focusing

SLAC NATIONAL ACCELERATOR LABORATORY

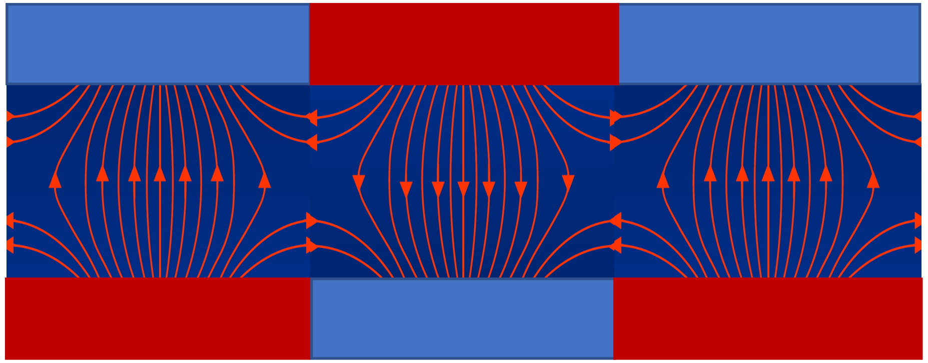
Los Alamos NATIONAL LABORATORY EST. 1943

LA-UR-21-20607

Planar Undulator



$\boldsymbol{B} = B_0 \hat{\boldsymbol{y}} \sin k_u z$



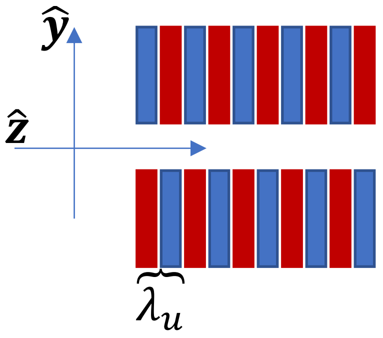
$\boldsymbol{B} = B_0 \hat{\boldsymbol{y}} \sin k_u z \cosh k_u y$
$+ B_0 \hat{\boldsymbol{z}} \cos k_u z \sinh k_u y$

- A previously assumed magnetic field does not satisfy $\nabla \cdot \boldsymbol{B} = 0$!
- The vertical sinusoidal field that satisfies Maxwell's equation has the vector potential $A_x = -\frac{B_0}{k_u} \cos k_u z \cosh k_u y$ that results in $B_z \neq 0$!
- Lorentz force (written as the second derivative with respect to z) results in:
  - $x'' = k_u \frac{K}{\gamma} \cosh k_u y \sin k_u z$, which is a standard FEL equation (2.7) for $y \approx 0$
  - $y'' + k_u \frac{K^2}{\gamma^2} \cos^2 k_u z \frac{\sinh 2k_u y}{2} = -p_x k_u \frac{K}{\gamma} \sinh k_u y \cos k_u z \approx 0$

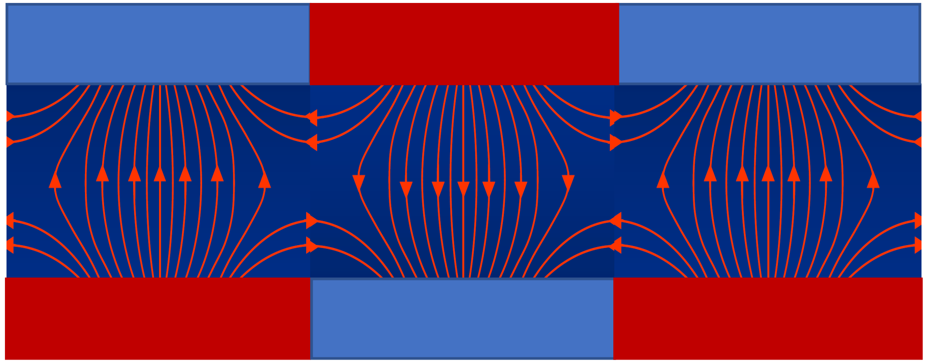
According to S.Y. Lee *Accelerator Physics,* 4th ed, chapter 4.III.2

LA-UR-21-20607

Planar Undulator



$\boldsymbol{B} = B_0 \hat{\boldsymbol{y}} \sin k_u z$



$\boldsymbol{B} = B_0 \hat{\boldsymbol{y}} \sin k_u z \cosh k_u y$
$+ B_0 \hat{\boldsymbol{z}} \cos k_u z \sinh k_u y$

- The nonlinear magnetic field can be neglected since $k_u y \ll 1$:
  - $y'' + \left(k_u^2 \frac{K^2}{\gamma^2} \cos^2 k_u z\right) y = 0$
- Many FEL codes assume that dynamics is ***averaged over the undulator period***:
  - $y'' = -\left\langle k_u^2 \frac{K^2}{\gamma^2} \cos^2 k_u z \right\rangle_u y = -k_u^2 \frac{K^2}{2\gamma^2} y$, which correspond to “natural focusing” of the undulator and the Hill’s equation $y'' + K_p y = 0$, with the focusing function $K_p = k_u^2 \frac{K^2}{2\gamma^2}$
- The helical undulator has $K_x = K_y = \frac{1}{\sqrt{2}} K_p$

SLAC NATIONAL ACCELERATOR LABORATORY

Los Alamos NATIONAL LABORATORY EST. 1943

LA-UR-21-20607

Matrix transformation

- Solution of the Hill's equation for the planar undulator is

$$\begin{bmatrix} y \\ y' \end{bmatrix}_{z\to z+L} = \begin{bmatrix} \cos\frac{K}{\sqrt{2}\gamma}k_uL & \frac{\sin\frac{K}{\sqrt{2}\gamma}k_uL}{\frac{K}{\sqrt{2}\gamma}k_u} \\ -\frac{K}{\sqrt{2}\gamma}k_u\sin\frac{K}{\sqrt{2}\gamma}k_uL & \cos\frac{K}{\sqrt{2}\gamma}k_uL \end{bmatrix} \begin{bmatrix} y \\ y' \end{bmatrix}_z$$

SLAC NATIONAL ACCELERATOR LABORATORY

Los Alamos NATIONAL LABORATORY EST. 1943

LA-UR-21-20607

Matrix transformation

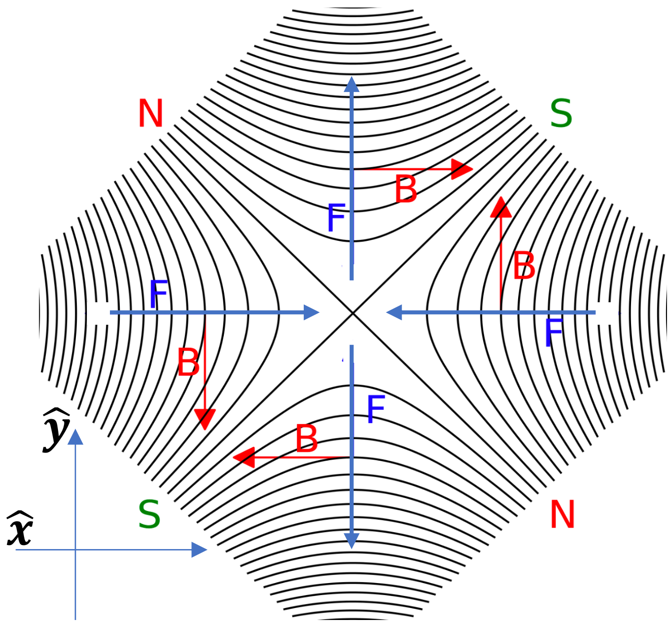
- Solution of the Hill's equation for the planar undulator is

$$\begin{bmatrix} y \\ y' \end{bmatrix}_{z\to z+L} \approx \begin{bmatrix} 1-\dfrac{K^2}{4\gamma^2}k_u^2L^2 & L \\ -\dfrac{K^2}{2\gamma^2}k_u^2L & 1-\dfrac{K^2}{4\gamma^2}k_u^2L^2 \end{bmatrix} \begin{bmatrix} y \\ y' \end{bmatrix}_z$$

- This matrix transformation corresponds to a thick lens of the thickness L and the focal length $f = \frac{2\gamma^2}{K^2k_u^2L} = \frac{6825[m^2]}{L[m]}$ for MaRIE-like x-ray FEL;
- The natural focusing is not sufficient for controlling transverse beam dynamics in FELs.

Strong focusing and FODO lattice

Focusing in Quadrupole



$\boldsymbol{B} = B_1(y\,\hat{\boldsymbol{x}} + x\,\hat{\boldsymbol{y}})$, where $B_1 = \frac{\partial B_y}{\partial x}\left[\frac{T}{m}\right]$ used by Genesis

in order to calculate the focusing strength: $k[m^{-2}] = 0.299\,\frac{B_1\left[\frac{T}{m}\right]}{E[GeV]}$

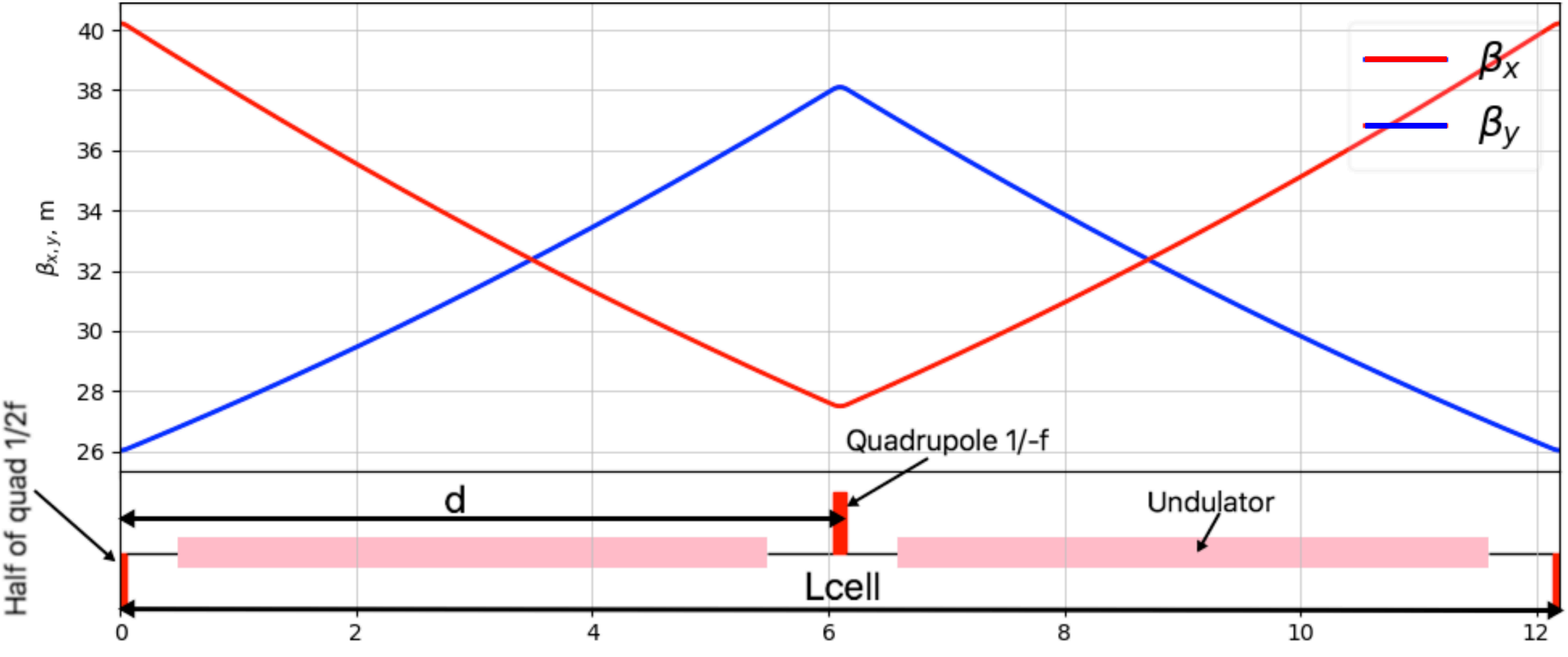
with a sign-convention $k > 0$ for horizontal focusing ('QUADF');

$$\begin{bmatrix} x \\ x' \end{bmatrix}_{z\to z+l} = \begin{bmatrix} \cos\sqrt{k}l & \frac{\sin\sqrt{k}l}{\sqrt{k}} \\ -\sqrt{k}\sin\sqrt{k}l & \cos\sqrt{k}l \end{bmatrix} \begin{bmatrix} x \\ x' \end{bmatrix}_z,$$ focusing plane

$$\begin{bmatrix} y \\ y' \end{bmatrix}_{z\to z+l} = \begin{bmatrix} \cosh\sqrt{k}l & \frac{\sinh\sqrt{k}l}{\sqrt{k}} \\ \sqrt{k}\sinh\sqrt{k}l & \cosh\sqrt{k}l \end{bmatrix} \begin{bmatrix} y \\ y' \end{bmatrix}_z,$$ defocusing plane

In the limit of $\sqrt{k}l \ll 1$, a thin quadrupole represents a thick lens of the length l and the focal length $f = (kl)^{-1}$.

FODO lattice – focusing in both planes



0.5F O D O 0.5F

FODO lattice in thin lens approximation

- A single FODO cell in matrix notations for x coordinate is

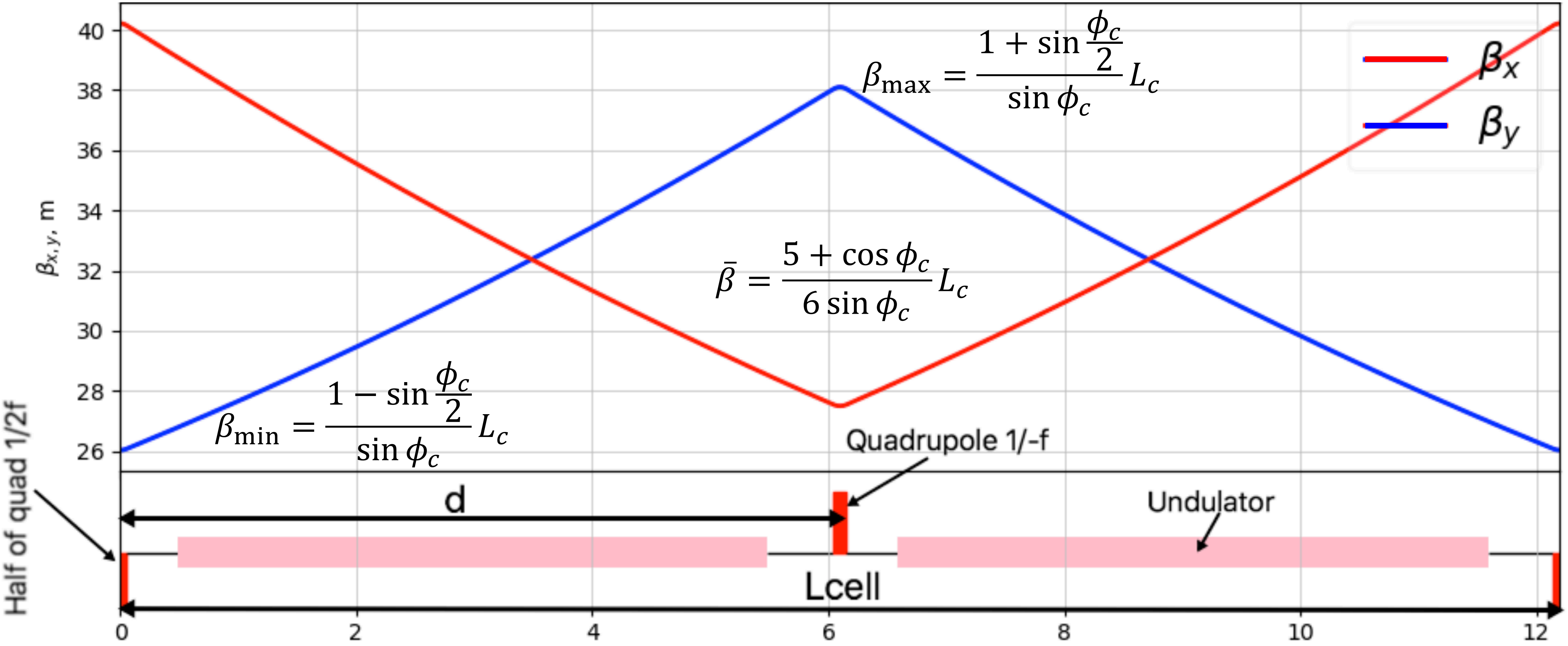
$$\begin{bmatrix} x \\ x' \end{bmatrix}_{z\to z+L_c} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2f} & 1 \end{bmatrix}\begin{bmatrix} 1 & \frac{L_c}{2} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ \frac{1}{f} & 1 \end{bmatrix}\begin{bmatrix} 1 & \frac{L_c}{2} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ -\frac{1}{2f} & 1 \end{bmatrix}\begin{bmatrix} x \\ x' \end{bmatrix}_z$$

that simplifies to

$$\begin{bmatrix} x \\ x' \end{bmatrix}_{z\to z+L_{cell}} = \begin{bmatrix} 1-\frac{L_c^2}{8f^2} & L_c\left(1+\frac{L_c}{4f}\right) \\ -\frac{L_c}{4f^2}\left(1-\frac{L_c}{4f}\right) & 1-\frac{L_c^2}{8f^2} \end{bmatrix} = U^T\begin{bmatrix} e^{i\phi_c} & 0 \\ 0 & e^{-i\phi_c} \end{bmatrix} U \begin{bmatrix} x \\ x' \end{bmatrix}_z$$

- The matrix transformation for y coordinate requires $f \to -f$ in the Eq. above;
- Phase advance is related to the transfer matrix by $\sin\frac{\phi_c}{2} = \frac{L_c}{4|f|}$, which results in a real phase and a stable solution if $|f| > L_c/4$.

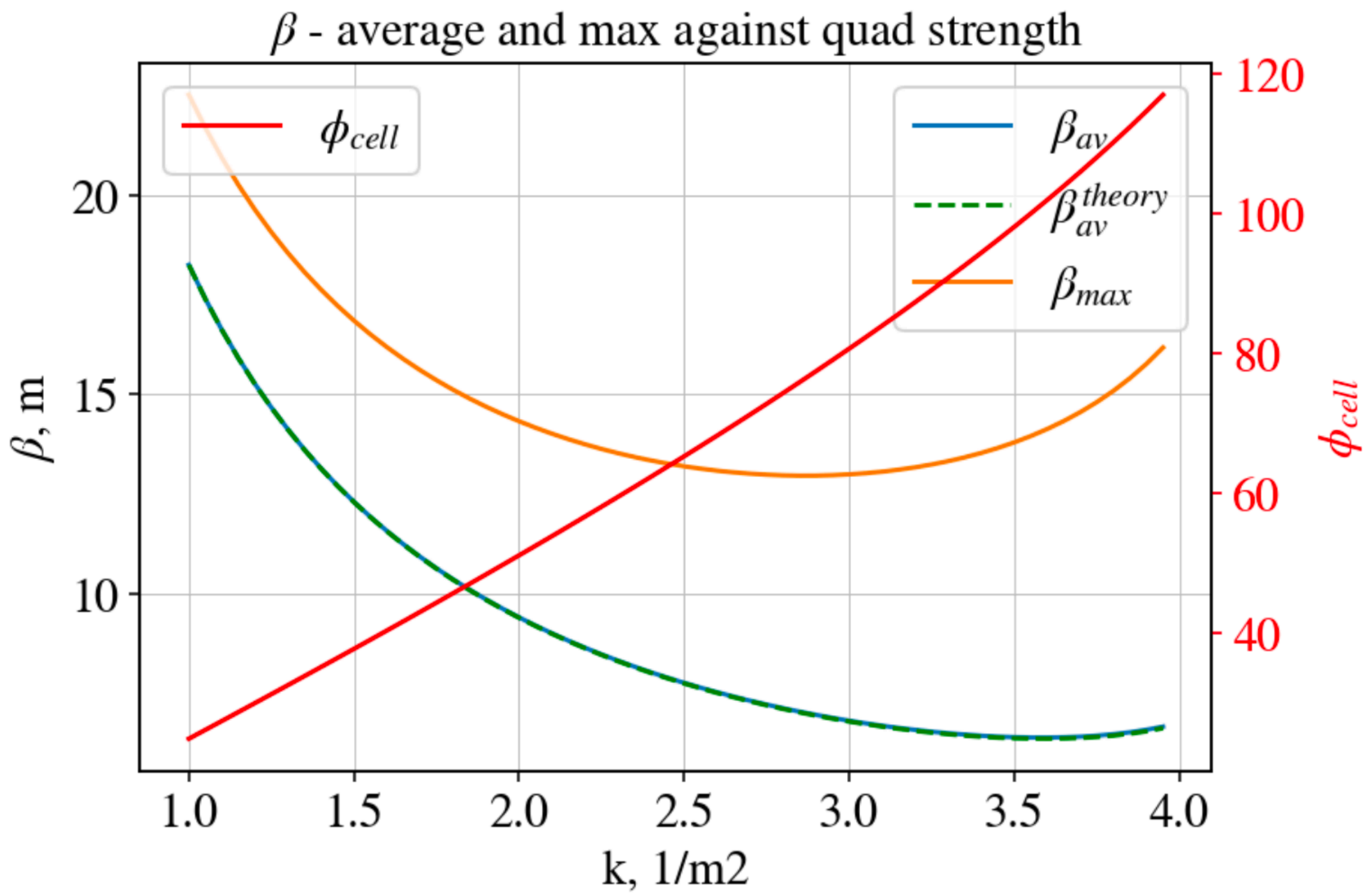
FODO lattice in thin lens approximation



The minimum possible average $\bar{\beta}_{min}$ gives us $\phi_{cell} = \pi - \arctan 2\sqrt{6} \approx 101^0$

LA-UR-21-20607

Beta function study for MaRIE-like x-ray FEL



Courant-Snyder Parametrization

- The most general form for matrix M_{FODO} with *unit modulus* can be parametrized as

$$M_{FODO} = \begin{bmatrix} \cos\phi_c + \alpha\sin\phi_c & \beta\sin\phi_c \\ -\gamma\sin\phi_c & \cos\phi_c - \alpha\sin\phi_c \end{bmatrix} = \mathbf{I}\cos\phi_c + \mathbf{J}\sin\phi_c\,,$$

where α, β and γ are Courant-Snyder parameters of the periodic solution, ϕ_c is the phase advance, $\mathbf{I}$ is the unit matrix, and

$$\mathbf{J} = \begin{bmatrix} \alpha & \beta \\ -\gamma & -\alpha \end{bmatrix},\ \text{with } \mathrm{Tr}(\mathbf{J}) = 0, \mathbf{J}^2 = -\mathbf{I} \text{ or } \beta\gamma = 1 + \alpha^2$$

- One can thus solve for FODO matched beam parameters;
- Using the property of matrix , we obtain the De Moivere's theorem:

$$M^n = \mathbf{I}\cos n\phi_c + \mathbf{J}\sin n\phi_c \quad \text{and} \quad M^{-1} = \mathbf{I}\cos\phi_c - \mathbf{J}\sin\phi_c$$

that describe an oscillation if the beam parameters are not FODO matched.

LA-UR-21-20607

Genesis Description

FODOlattice.ipynb Input parameters

```
# UNDULATOR parameters
XLAMD = 0.0186  # undulator wavelength
ku = 2*np.pi/XLAMD
AW0 = 0.86  # rms Undulator parameter
K = np.sqrt(2)*AW0
# FODO parameters
F1ST = 5
QUADF = 30
FL = 10
QUADD = 30
FD = 10

DRL = 100
# Beam parameters
GAMMA0 = 12e9/0.511e6  # beam energy in mc2
```

http://genesis.web.psi.ch/Manual/parameter_focusing.html
http://genesis.web.psi.ch/Manual/parameter_undulator.html
http://genesis.web.psi.ch/Manual/parameter_beam.html

Derived parameters

```
# Focusing quads
lf1st = F1ST*XLAMD
lf = FL*XLAMD
kf = 585*QUADF/GAMMA0
# Defocusing quads
ld = FD*XLAMD
kd = -585*QUADD/GAMMA0
# Undulator focusing
L = DRL*XLAMD
Ku = K**2*ku**2/(2*GAMMA0**2)
```

```
# Thin lens approximation
f = (lf*kf)**-1
phi_c = 2*np.arcsin((lf+ld+2*L)/(4*f))
print(f"Simple phi_c={phi_c}")
print(f"Simple beta_max={(lf+ld+2*L)*(1+np.sin(phi_c/2))/np.sin(phi_c)}")
print(f"Simple beta_min={(lf+ld+2*L)*(1-np.sin(phi_c/2))/np.sin(phi_c)}")
```

```
Simple phi_c=0.2859559977103042
Simple beta_max=16.57388512332508
Simple beta_min=12.43970009250435
```

SLAC NATIONAL ACCELERATOR LABORATORY

Los Alamos NATIONAL LABORATORY EST. 1943

LA-UR-21-20607

Horizontal FODO in a thick lens approximation

```
FODOx = focus(kf, lf1st) @ \
        drift(L) @ focus(kd, ld) @ drift(L) @ \
        focus(kf, lf1st)
print(FODOx)
```

```
[[ 0.96200859  4.64624157]
 [-0.01604296  0.96200859]]
```

```
phi_c = (np.arccos(np.trace(FODOx)/2))
print(phi_c)
```

```
0.27653006629258703
```

```
beta_x = FODOx[0,1]/np.sin(phi_c)
print(beta_x)
```

```
17.018003459945035
```

```
np.sqrt(beta_x*0.2e-6/GAMMA0)
```

```
1.2038964357474105e-05
```

- The phase advance is now smaller;
- The maximum of the beta function is now greater.

Vertical FODO

Without “natural focusing”

```
FODOy = focus(-kf, lf1st) @ \
        drift(L) @ focus(-kd, ld) @ drift(L) @ \
        focus(-kf, lf1st)
print(FODOy)
```

```
[[ 0.96200859  3.53536912]
 [-0.02108393  0.96200859]]
```

```
phi_c = (np.arccos(np.trace(FODOy)/2))
print(phi_c)
```

```
0.2765300662925874
```

```
beta_y = FODOy[0,1]/np.sin(phi_c)
print(beta_y)
```

```
12.949159647934803
```

```
np.sqrt(beta_y*0.2e-6/GAMMA0)
```

```
1.0501603512555186e-05
```

With “natural focusing”

```
FODOy = focus(-kf, lf1st) @ \
        focus(Ku, L) @ focus(-kd, ld) @ focus(Ku, L) @ \
        focus(-kf, lf1st)
print(FODOy)
```

```
[[ 0.96085158  3.53385877]
 [-0.0217225   0.96085158]]
```

```
phi_c = (np.arccos(np.trace(FODOy)/2))
print(phi_c)
```

```
0.2807367077713572
```

```
beta_y = FODOy[0,1]/np.sin(phi_c)
print(beta_y)
```

```
12.754684818958047
```

```
np.sqrt(beta_y*0.2e-6/GAMMA0)
```

```
1.042244688359981e-05
```

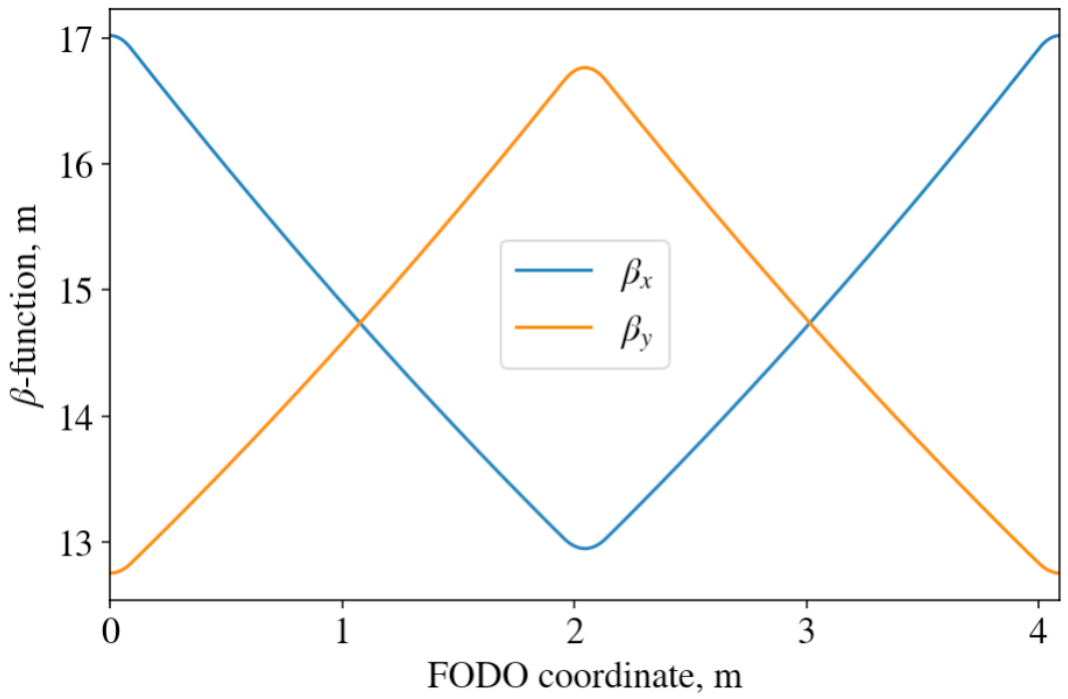
Betatron Envelope Equation

- The focusing function $K(s)$ is real and hence the amplitude and phase functions satisfy $w'' + K\,w \;-\frac{1}{w^3} \;=\; 0$, and $\psi' = \frac{1}{w^2}$, where the normalization is chosen such that w^2 is exactly the Courant-Snyder β-function and the Courant-Snyder function $\alpha = -\frac{\beta'}{2} = -w\,w'$.
- We want to use the numerical formalism in order to calculate the β-function in the FODO cell instead of a single point value obtained by the matrix formalism;
- We then use the numerical formalism in order to evaluate the average β-function in the FODO cell and compare this value to arithmetic average between max and min values of the β-function:

$$\bar{\beta} = \frac{\beta_{max}+\beta_{min}}{2} \quad \text{vs} \quad \bar{\beta} = \frac{1}{L_c}\int_0^{L_c} \beta(s)ds$$

Betatron envelope equation results

BetatronEnvelopeEquation.ipynb



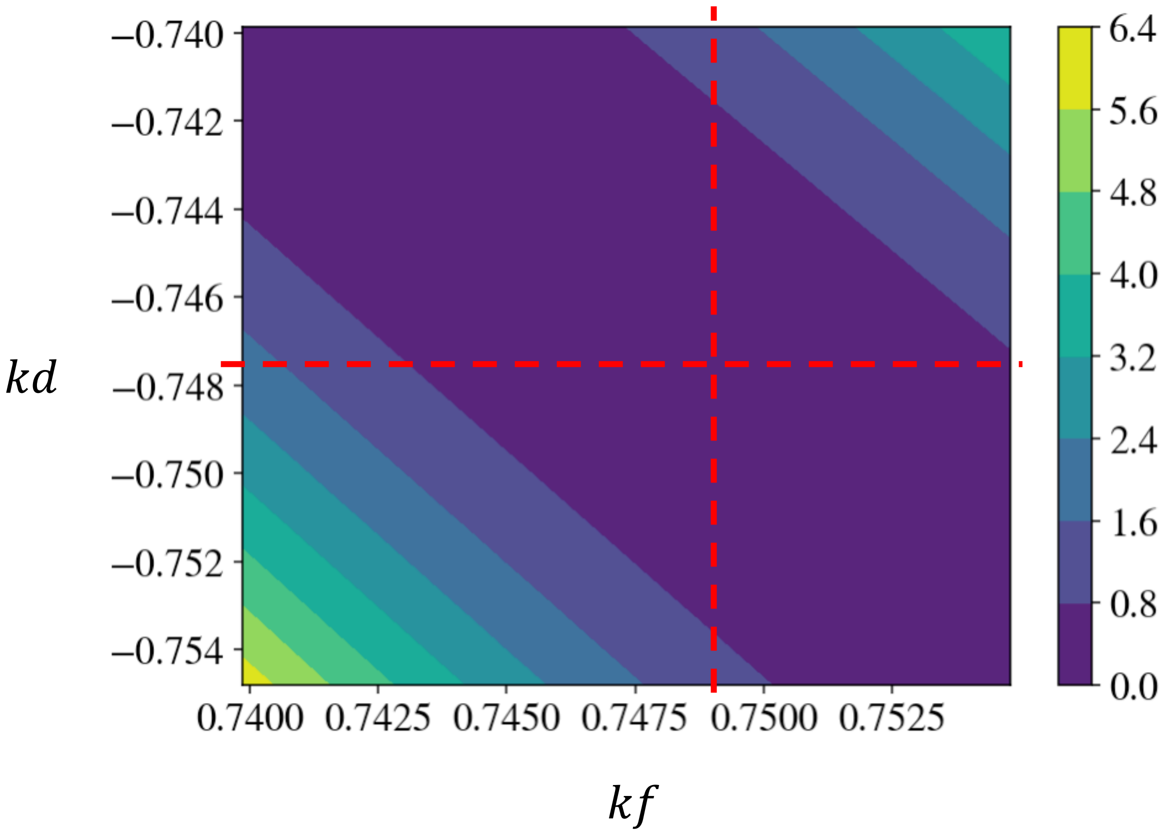
$\bar{\beta}_{\mathrm{x}} = 14.898$ m and $\bar{\beta}_{\mathrm{y}} = 14.675$m are both less than the arithmetically average $\bar{\beta}$-function.

- The thick lens solution has smooth turns inside the quadrupoles instead of sharp ones in the thin lens approximation;
- The vertical solution is focused more due to the natural focusing in the undulator;
  - Horizontal solution:
    - The maximum x beta function 17.018 m.
    - Phase advance is 0.27653 rad.
    - The minimum x beta function 12.949 m.
  - Vertical solution:
    - The maximum y beta function 12.755 m.
    - Phase advance is 0. 28074 rad.
    - The minimum y beta function 16.763 m.

LA-UR-21-20607

$\bar{\beta}$-function matching

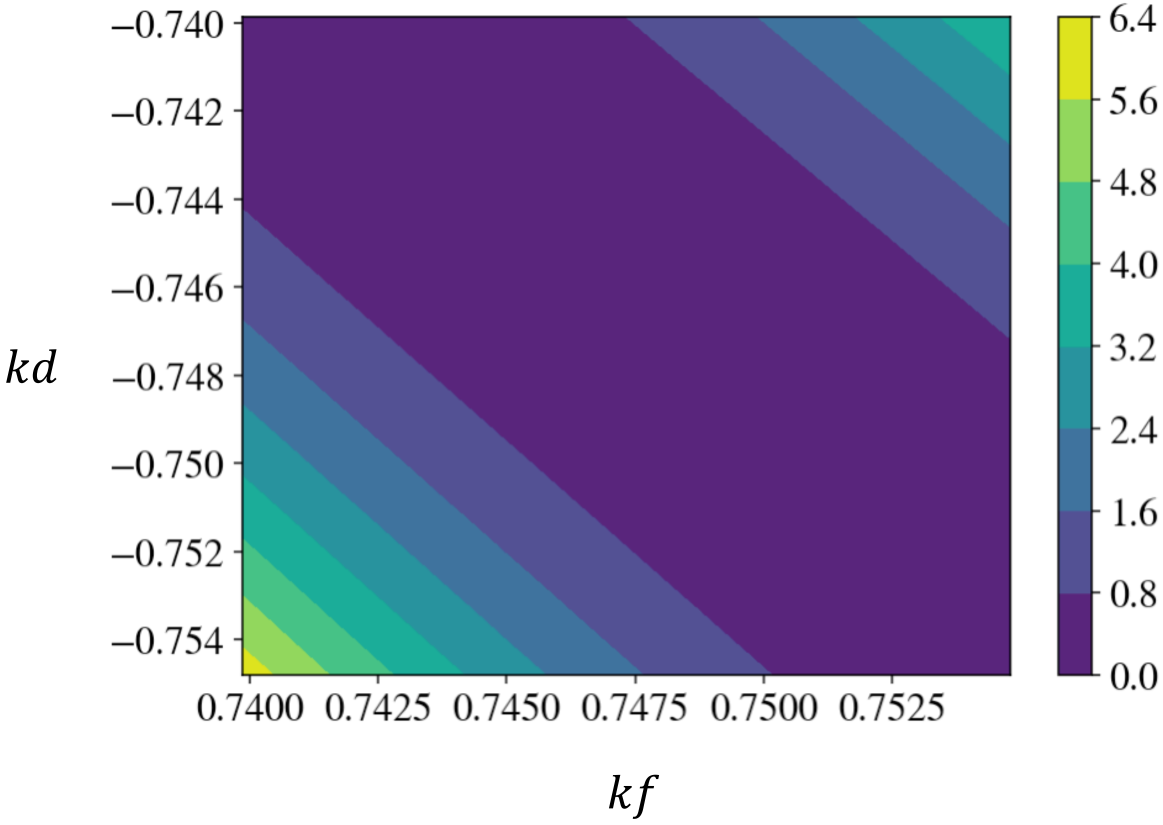
β-function mismatch $\left(\bar{\beta}_x - \bar{\beta}_y\right)^2$



- The matched solution requires a focusing quad, which defocuses in the vertical plane, to be stronger than a defocusing quad in order to compensate for the natural focusing of the undulator:
  - QUADF = 30.061 T/m;
  - QUADD = -30 T/m;

$\bar{\beta}$-function matching

$\boldsymbol{\beta}$-function mismatch $(\bar{\beta}_x - \bar{\beta}_y)^2$



- Horizontal solution:
  - The phase advance is 0.27892 rad.
  - Average x beta function is 14.771 m.
  - The maximum x beta function is 16.876 m.
  - The minimum x beta function is 12.837 m.
- Vertical solution:
  - The phase advance is 0.27892 rad.
  - Average y beta function is 14.771 m.
  - The maximum y beta function is 16.874 m.
  - The minimum y beta function is 12.836 m.
- The $\bar{\beta}$-functions are matched but the maximum and minimum are not the same due to the natural focusing in one plane modifying the beta function evolution.

Ming Xie analysis of 3D FELs

SLAC NATIONAL ACCELERATOR LABORATORY

Los Alamos NATIONAL LABORATORY EST.1943

LA-UR-21-20607

Ming Xie solution

- Ming Xie solution has been implemented in zfel.old_scripts.mingxie()
- Inputs keyword arguments:

  sigma_x # RMS beam size

  und_lambda # Undulator period (m)

  und_k # Undulator K

  current # Beam current (A)

  gamma # Relativistic gamma

  norm_emit # Normalized emittance (m-rad)

  sigma_E # RMS energy spread (eV)
- Output as dictionary:

  gain_length # Gain length (m

  saturation_length # Saturation length (m)

  saturation_power # Saturation power (W)

  fel_wavelength # FEL wavelength (m)

  pierce_parameter # Pierce parameter (rho)
- My Mathematica implementation is presented here.

```
SaturationL[ση_, ε_, β_, λu_:1.86×10^-2, λ1_:0.3×10^-10, bEnergy_:12×10^9, Ib_:3000] :=
 Block[{ku, k1, γb, K, ξ, JJ, σr, IA = 17050, ρ, LG0, ηγ, ηε, ηd, Λ, Pb, Psat, α, Pn},
  (*Design Optimization for an X-ray Free Elecgron Laser Driven by SLAC Linac
   Ming Xie,Lawrence Berkeley Laboratory,Berkeley,CA 94720,USA
   http://accelconf.web.cern.ch/AccelConf/p95/ARTICLES/TPG/TPG10.PDF*)
  ku = 2 π / λu;   k1 = 2 π / λ1;
  γb = 1 + bEnergy / 510999.;
  K = √2 √(2 γb^2 λ1 / λu - 1);
  ξ = K^2/(4 + 2 K^2);   JJ = BesselJ[0, ξ] - BesselJ[1, ξ];
  σr = √(ε β / γb);
  ρ = (1/16 Ib/IA K^2 JJ^2/(γb^3 σr^2 ku^2))^(1/3); (*FEL parameter*)
  LG0 = λu/(4 π √3 ρ); (*1D Gain Length*)
  (*Ming Xie Parametrizaton*)
  ηd = LG0/(2 k1 σr^2);   ηε = LG0 ((4 π) ε / γb)/(β λ1);   ηγ = 4 π LG0/λu ση;
  Λ = 0.45 ηd^0.57 + 0.55 ηε^1.6 + 3 ηγ^2 + 0.35 ηε^2.9 ηγ^2.4 + 51 ηd^0.95 ηγ^3 + 5.4 ηd^0.7 ηε^1.9 + 1140 ηd^2.2 ηε^2.9 ηγ^3.2;
  (*Conclusions*)
  Pb = Ib bEnergy;
  Psat = 1.6 ρ Pb / (1 + Λ)^2;
  α = 1 / 9;
  Pn = ρ^2 bEnergy 1.60217662×10^-19 × 299792458 / λ1;
  Return[LG0 (1 + Λ) Log[Psat / Pn / α]];
 ];
```

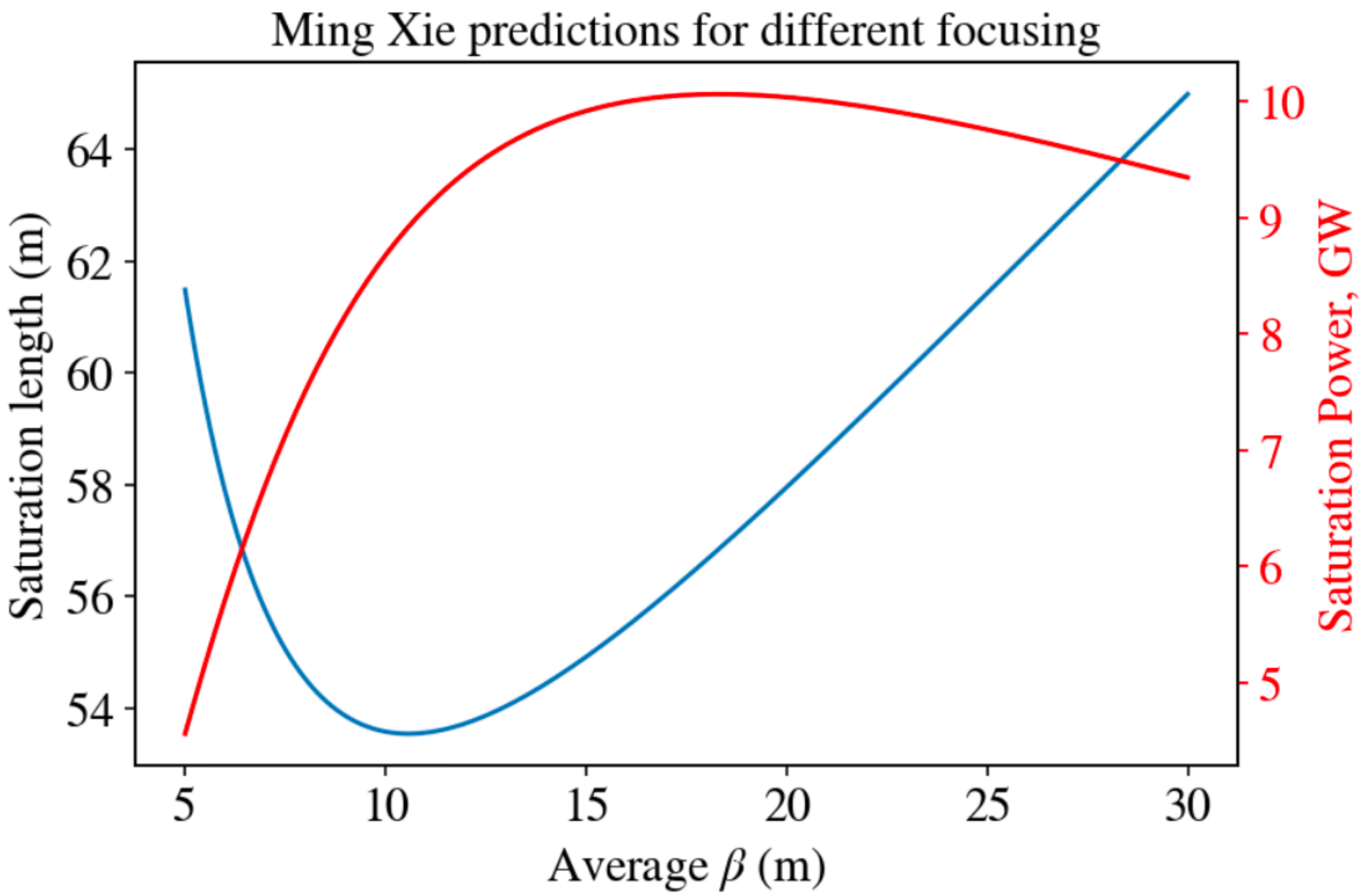
Ming Xie evaluation – single value

```
XLAMD = 1.86e-2  # undulator period
XLAMDS = 0.3e-10  # desired x-ray wavelength
CURPEAK = 3e3  # peak current
GAMMA0 = 12e9/0.511e6  # energy in mc2
energy = 0.511e6*GAMMA0  # beam energy in eV
DELGAM = 1.5e-4
sigma_e = DELGAM*energy  # energy spread in eV
AWO = np.sqrt(2*GAMMA0**2*XLAMDS/XLAMD-1)  # we use resonant condition formula here
K = np.sqrt(2)*AWO
EMITX = 0.2e-6  # normalized emittance in x
beta = 15  # beta function in meters
RXBEAM = np.sqrt(beta*EMITX/GAMMA0)  # the corresponding beam size
```

```
# Some parameters
params = {
    'sigma_x':RXBEAM,
    'und_lambda':XLAMD,
    'und_k':K,
    'current':CURPEAK,
    'gamma':GAMMA0,
    'norm_emit':EMITX,
    'sigma_E':sigma_e}


mingxie(**params)
```

{'gain_length': 3.21,
'saturation_length': 54.91,
'saturation_power’: 9.916e9,
'fel_wavelength': 3e-11,
'pierce_parameter': 4.125e-4}

Ming Xie evaluation – scan



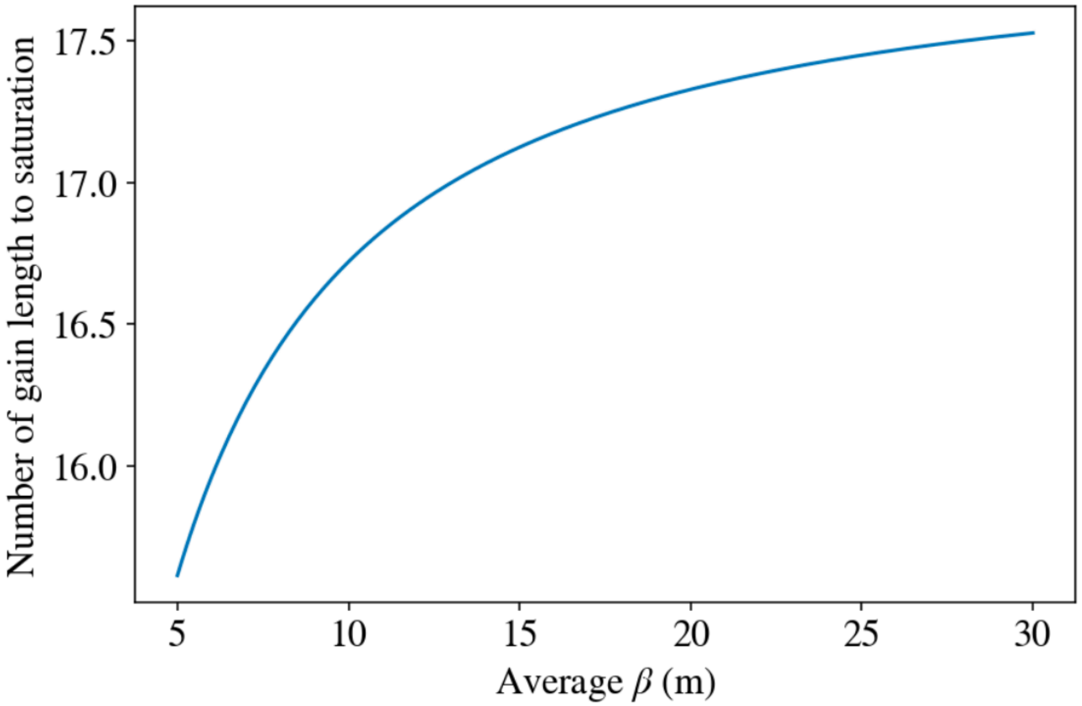
- We can use Ming Xie predictions in order to find the optimum beam size for MaRIE-like x-ray FEL;
- We can see that, although $\bar{\beta} = 10$ m results in a shorter saturation length (higher gain), the highest saturated power is reached at $\bar{\beta} = 15\ m$;
- Picking up $\bar{\beta}$ provides guidance on the FODO lattice design, which we discussed in the previous section;
- Please recall the value of the previously discussed $\bar{\beta}$ values; they were closer to the power maximum than to the gain maximum, which is inversely related to the saturation length.

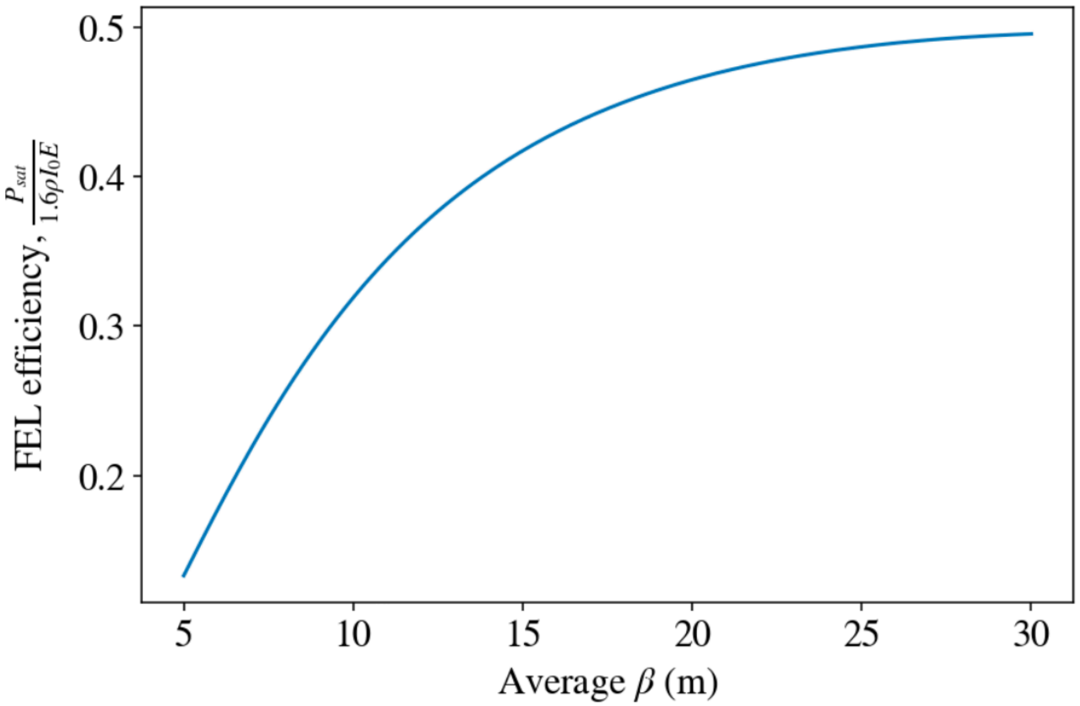
SLAC NATIONAL ACCELERATOR LABORATORY

Los Alamos NATIONAL LABORATORY EST. 1943

LA-UR-21-20607

Ming Xie evaluation – scan

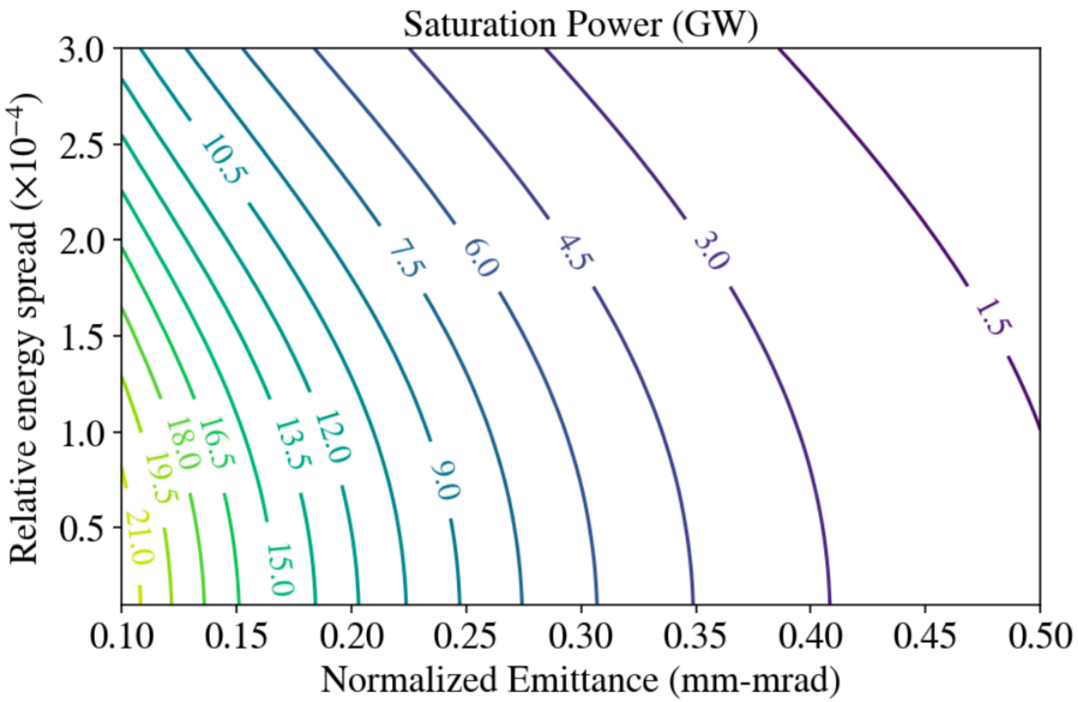


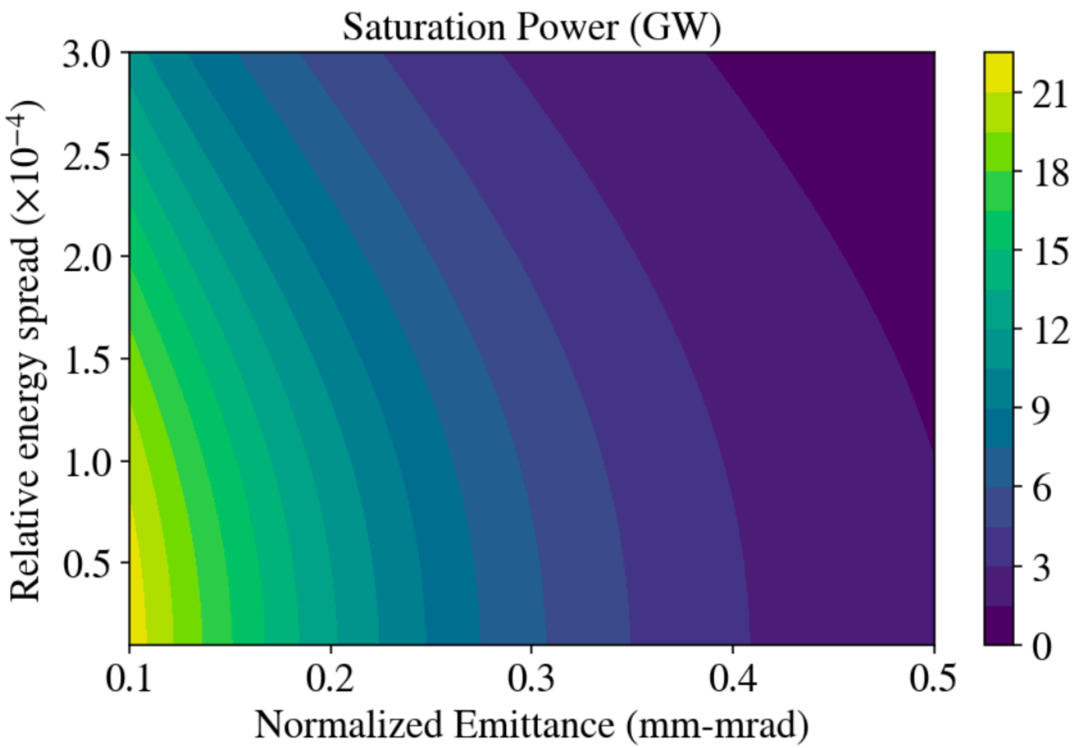


- The saturation length measured in the number of gain length is less than predicted by 1D theory;
- The saturated power is less than 50% of that predicted by 1D theory;
- These degradations are due to the 3D effects.

ZFEL execution – 2 variable scan

LA-UR-21-20607





- $\bar{\beta} = 15\ m$ case is studied here;
- This analysis shows that XFEL performance is highly sensitive to emittance and energy spread;
- The studied design here is emittance dominated and one would require a low emittance beam!

3D Numerical simulator – LUME-Genesis

LA-UR-21-20607

Genesis v2

- We will focus on Genesis v2, which is a stable version of a very popular FEL modeling code (http://genesis.web.psi.ch/) written in Fortran;
- It uses undulator averaged approximation and thus expresses all the distances in the units of undulator period XLAMD;
- Electromagnetic fields are expressed on the Cartesian grid;
- Electrons are represented by an equal number of macroparticles arranged in slices, one resonant wavelength XLAMDS long;
- Slices are ZSEP wavelength apart;
- LUME-genesis is python interface to setup, run and analyze Genesis v2 simulations;
- Genesis v4, rewritten in C, is under active development now (https://github.com/svenreiche/Genesis-1.3-Version4);

SLAC NATIONAL ACCELERATOR LABORATORY

Los Alamos NATIONAL LABORATORY EST.1943

LA-UR-21-20607

Configuring Genesis executable

- Genesis v2 is written in Fortran and does not support dynamic array allocations;
- Maximum array sizes have to be provided prior to compilation of an executable;
- ‘genesis.def’ file contains these variables https://github.com/slaclab/Genesis-1.3-Version2/blob/master/genesis.def , which is a modification of the original Genesis source code.

```
      parameter(
     +          genver = 2.3,               !genesis version
#if WIN == 1
     +                  platf  ='Win ',             !platform
#else
     +                  platf  ='Unix',             !platform
#endif
     +          original = 1 ,              !indicator for original filetype
     +          f_sdds = 2 ,                !indicator for sdds filetype
     +          npmax  = 1000001,            !# of particles
     +          nzmax  = 10000,             !# of integration steps
#if WIN == 1
     +          nsmax  = 50000,             !# of slices
     +          nhmax  = 7,                 !maximum of harmonics
#else
     +          nsmax  = 150000,             !# of slices
     +          nhmax  = 5,                 !maximum of harmonics
```

```
#endif
     +          ndmax   = 1250000,          !maximum of particle in imported distribution
     +          keepdist = 1,               !<> 0 keeps distribution in memory
     +          eev     = 510999.06d0,      !energy units (mc^2) in ev
     +          vacimp = 376.73d0,          !vacuum impedence in ohms
     +          ce      = 4.803302d-11,     !speed of light * electron charge
     +          pi      = 3.14159265358979d0, !pi
     +          pihalf = pi/2.d0,           !pi/2
     +          twopi   = 2.d0*pi,          !2*pi
     +          tiny    = 1.d-25,           !check i for precission
     +          small   = 1.d-7,            !check ii for precission
     +          nrgrid = 1000,              !number of radial points for s. c.
#if WIN == 1
     +          ncmax   = 261)              !# of gridpoints of cartesian mesh
#else
     +          ncmax   = 503)              !# of gridpoints of cartesian mesh
#endif
```

NATIONAL ACCELERATOR LABORATORY

Los Alamos NATIONAL LABORATORY EST. 1943

LA-UR-21-20607

Genesis v2: Input file http://genesis.web.psi.ch/Manual/files1.html

- Concise description of the input parameters could be found in https://github.com/slaclab/Genesis-1.3-Version2/blob/master/input.f#L1272 as well as https://github.com/ocelot-collab/ocelot/blob/master/ocelot/adaptors/genesis.py#L202
- Genesis generates 'template.in' if ran without an input file provided!

Congratulations!

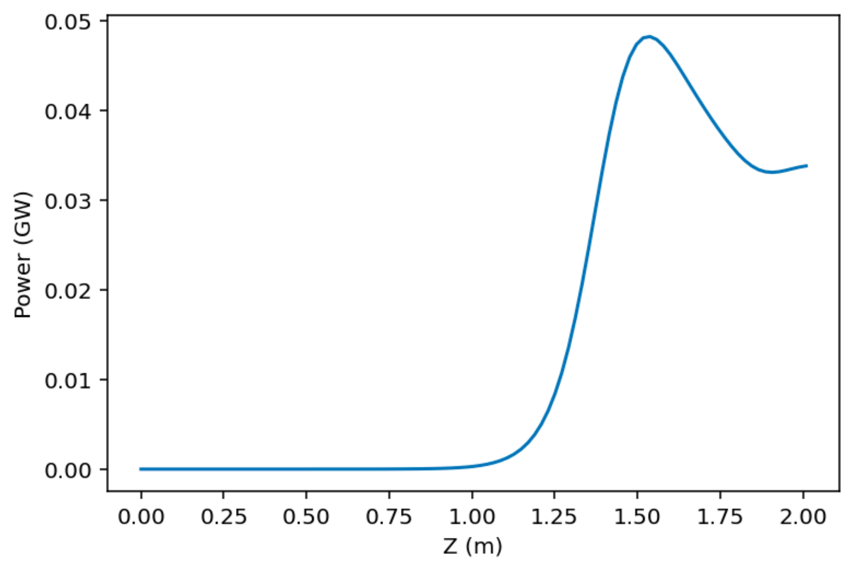
This is your first Genesis simulation!

```
!genesis2
```

```
--------------------------------
Genesis 1.3 has begun execution
(Version 2.3 Unix)

Please enter input file name
***  File-error:
***  cannot be opened
***  creating template file: template.in
***  closing files

Genesis run has finished
-------------------------
```

```
from genesis import Genesis
gen = Genesis('template.in')
gen.run()
```

```
plot as plt
['data']['z'], gen.output['data']['power'][0
 plt.ylabel('Power (GW)');
```



SLAC NATIONAL ACCELERATOR LABORATORY

Los Alamos NATIONAL LABORATORY EST.1943

LA-UR-21-20607

Genesis v2: MaRIE input file

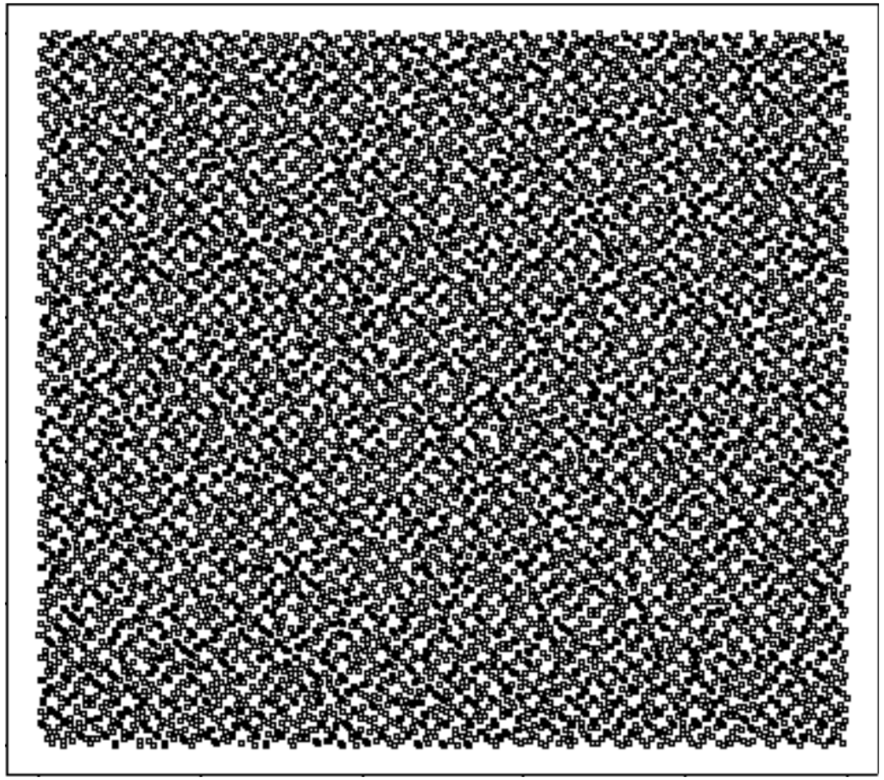
```
genesis_bin='/home/vagrant/.local/bin/genesis2-mpi'
gen = Genesis('template.in', genesis_bin=genesis_bin)
gen.binary_prefixes = ['mpirun', '-n', '4']

# undulator
gen['xlamd'] = 0.0186  # undulator wavelength, m
gen['aw0'] = gen['awd'] = 0.86  # rms undulator parameter
gen['nwig'] = int(80/gen['xlamd'])  # undulator length in xlamd

# focusing
gen['f1st'] = 5  # half F length in FODO measured in xlamd
gen['fl'] = 10   # full F length in FODO measured in xlamd
gen['quadf'] = 30  # focusing in x quadrupole gradient, T/m
gen['dl'] = 10   # full D length in FODO measured in xlamd
gen['quadd'] = 30  # defocusing in x quadrupole gradient, T/m
gen['drl'] = 100 # full O length in FODO measured in xlamd

# electron beam
gen['curpeak'] = 3000  # current, A
gen['curlen'] = 0  # negative for flattop; positive for Gaussian
gen['gamma0'] = 12e9/0.511e6  # beam energy, mc^2
gen['delgam'] = 1.5e-4*gen['gamma0']  # relative energy spread
gen['rxbeam'] = 1.2038964357474105e-05  # rms size, m
gen['rybeam'] = 1.0422446883599981e-05  # rms size, m
gen['emitx'] = gen['emity'] = 0.2e-6  # normalized emittance, m rad
gen['npart'] = 2**10  # number of macroparticles in a bucket
```

```
# radiation at resonant condition
gen['xlamds'] = gen['xlamd']*(1+gen['aw0']**2)/(2*gen['gamma0']**2)
gen['prad0'] = 1e4  # shot noise power, W
gen['zrayl'] = 24  # Rayleigh length, m
gen['zwaist'] = 0  # focul point location, m

# mesh
gen['ncar'] = 151  # number of mesh points, ODD is advised
gen['dgrid'] = 100e-6  # [-dgrid, dgrid], m

# simulation
gen['delz'] = 1  # integration step measured in xlamd
gen.run()
gen.output['run_info']
```

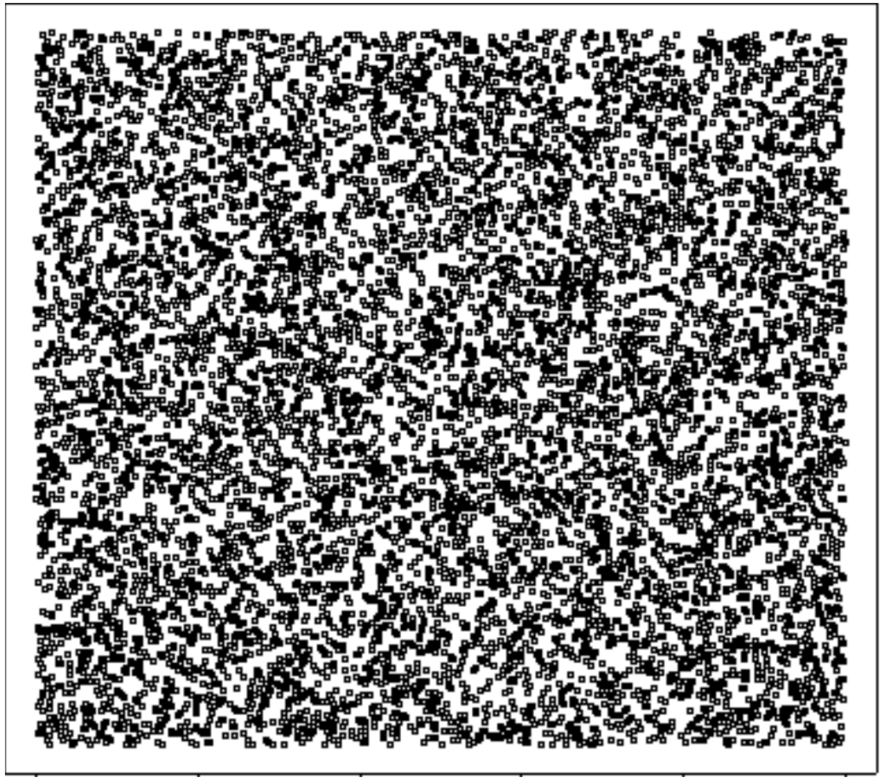
```
{'start_time': 1611354311.8145232,
 'run_script': 'mpirun -n 4 /home/vagrant/.local/bin/genesis2-mpi ge
nesis.in',
 'run_time': 6.585921049118042,
 'run_error': False}
```

Particle loading – ‘quiet start’

The first 10000 points in the same sequence. These 10000 comprise the first 1000, with 9000 more points.



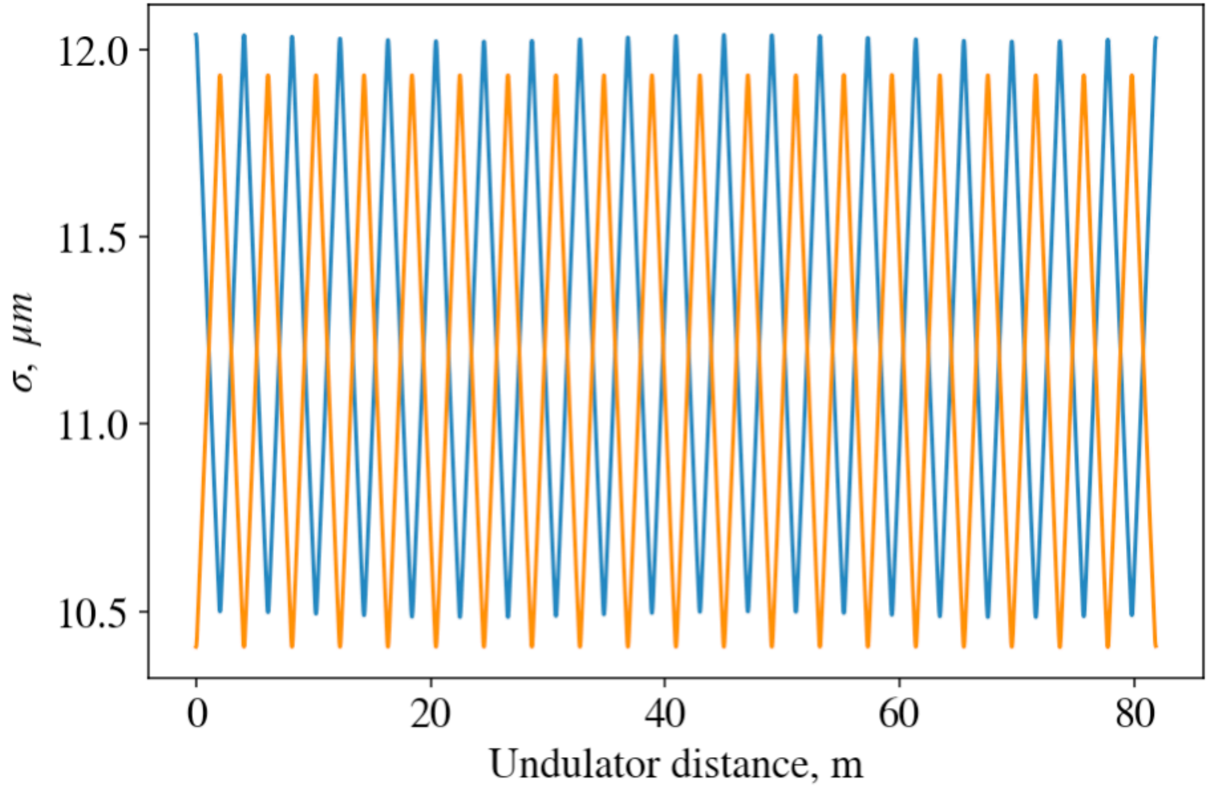
The first 10000 points in a sequence of uniformly distributed pseudorandom numbers. Regions of higher and lower density are evident.

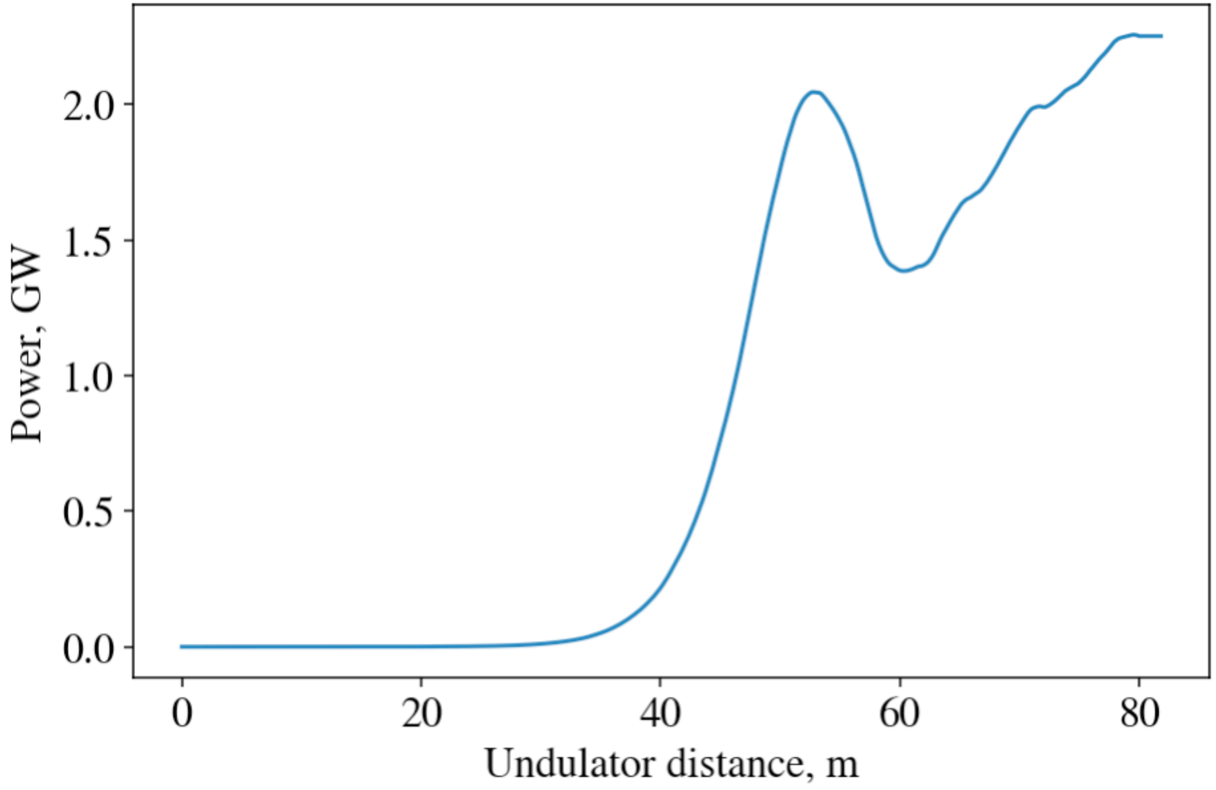


Genesis uses Hammersley sequence for quiet start instead of pseudorandom numbers that introduces bunching!
https://en.wikipedia.org/wiki/Low-discrepancy_sequence#Hammersley_set

Results of the no SASE simulation

LA-UR-21-20607

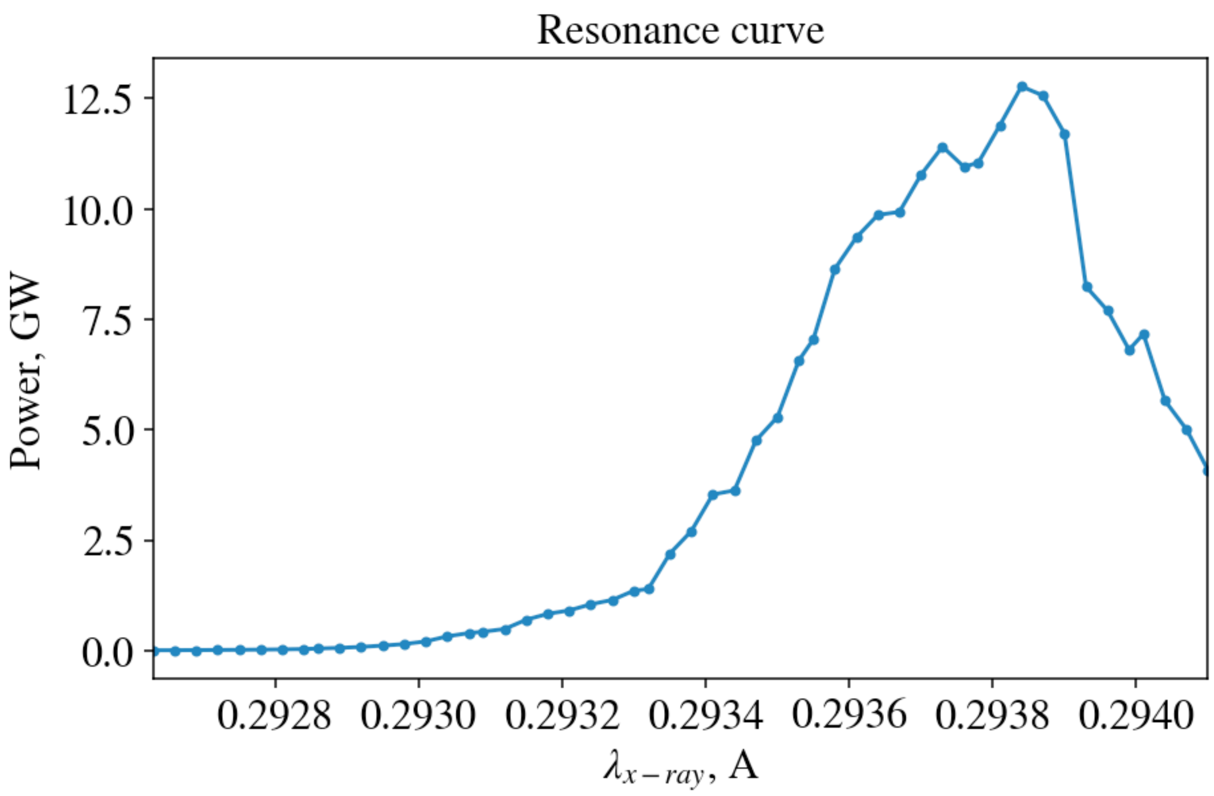




Electron beam size is FODO matched but not $\bar{\beta}$ matched.

Generated power is less than expected from Ming Xie. Why?

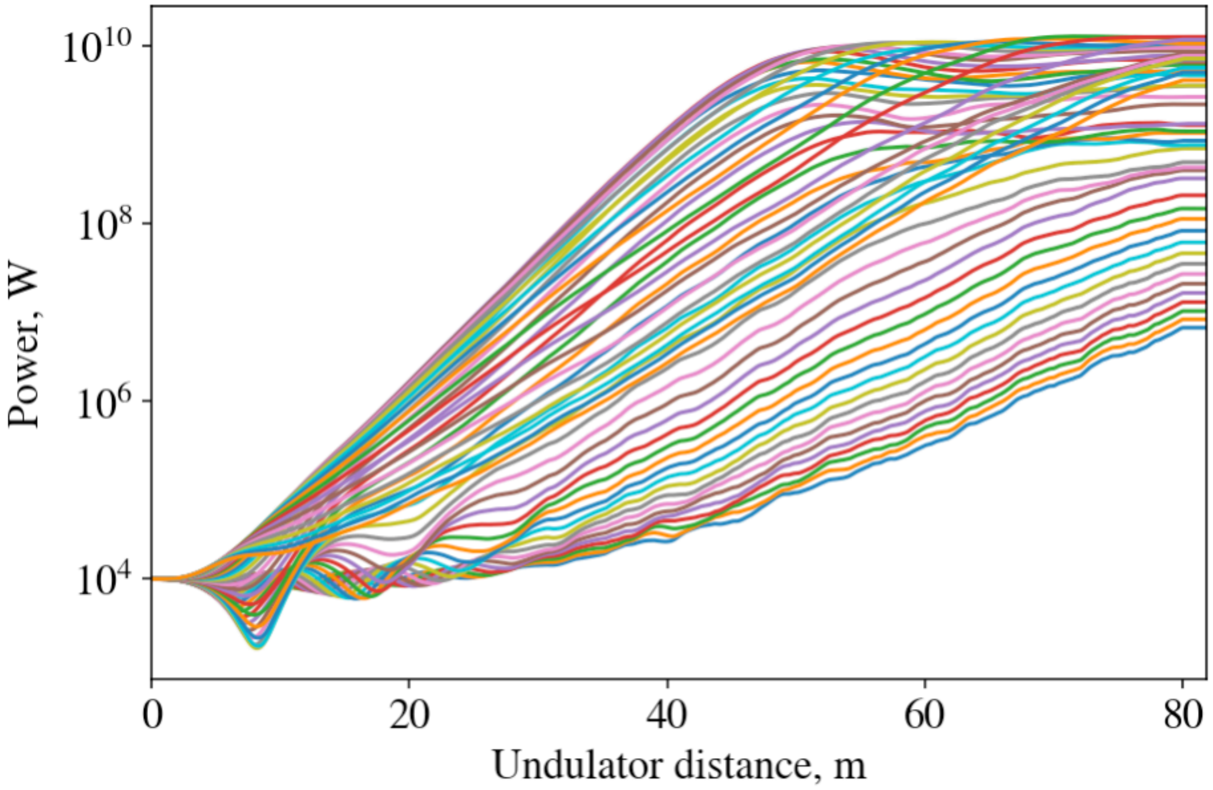
Power optimization

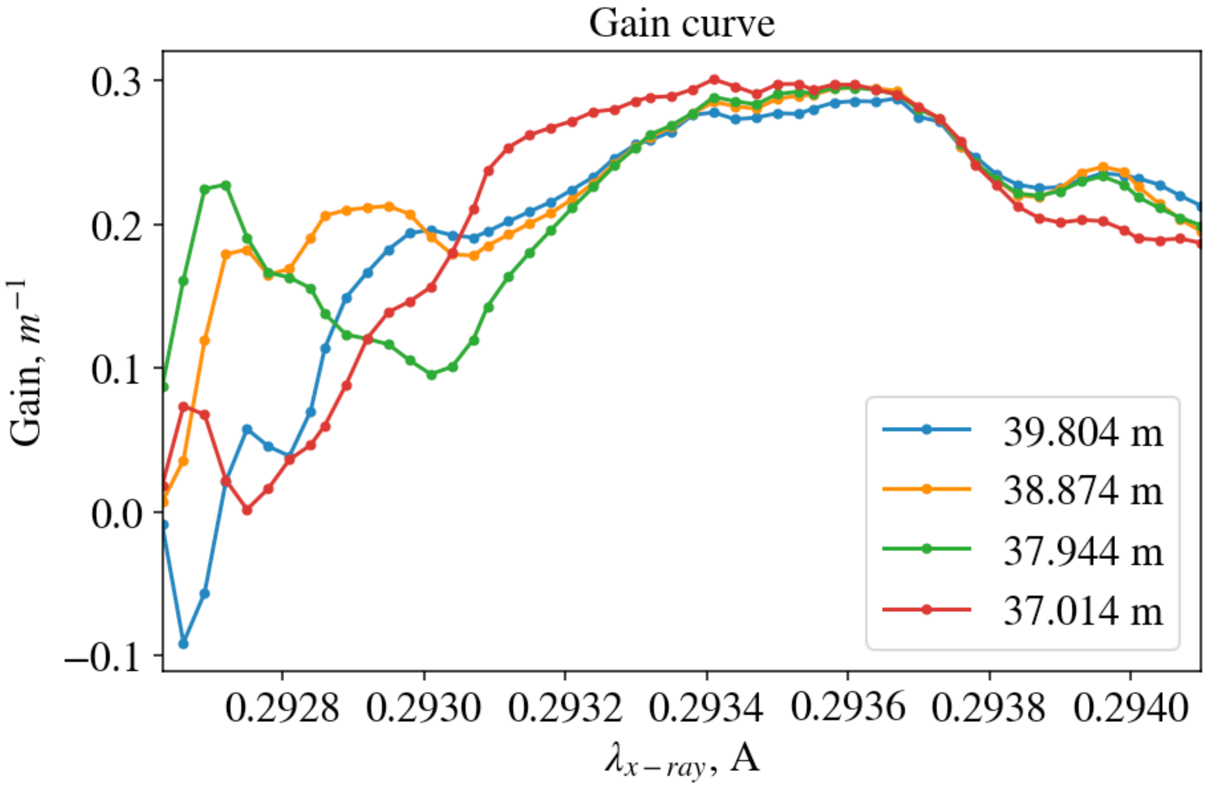



```
gen['iscan'] = 4
gen['nscan'] = 52
gen['svar'] = 0.0025
gen.run()
gen.output['run_info']
```

- The steady state simulation keeps the radiation wavelength fixed in order to determine the size of the slice;
- This type of simulation assumes no slippage of the radiation with respect to the electrons;
- Any electrons that fall behind get reintroduced into the slice assuming that an identical slice exists ahead of the simulated slice;
- The expected wavelength was 2.9337e-11 m but we have found 2.9384e-11 m, which is 0.0016 relative shift.

Gain optimization

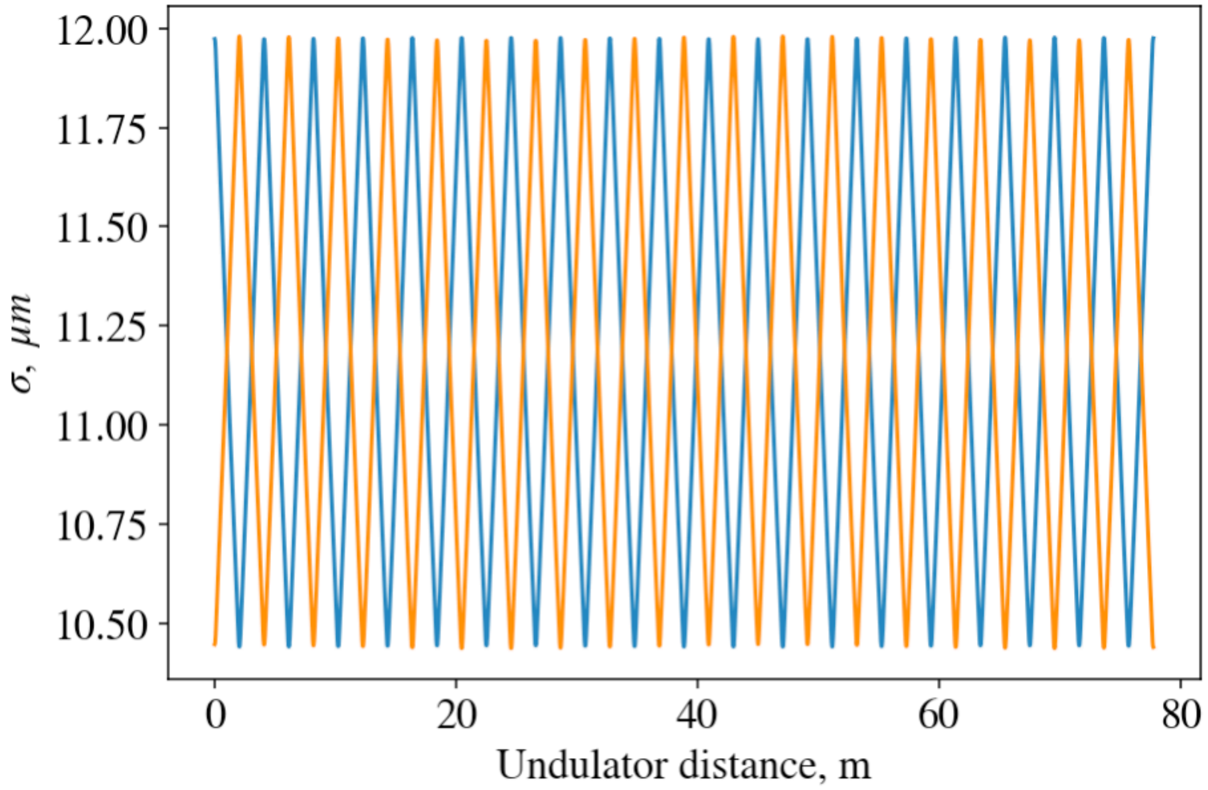


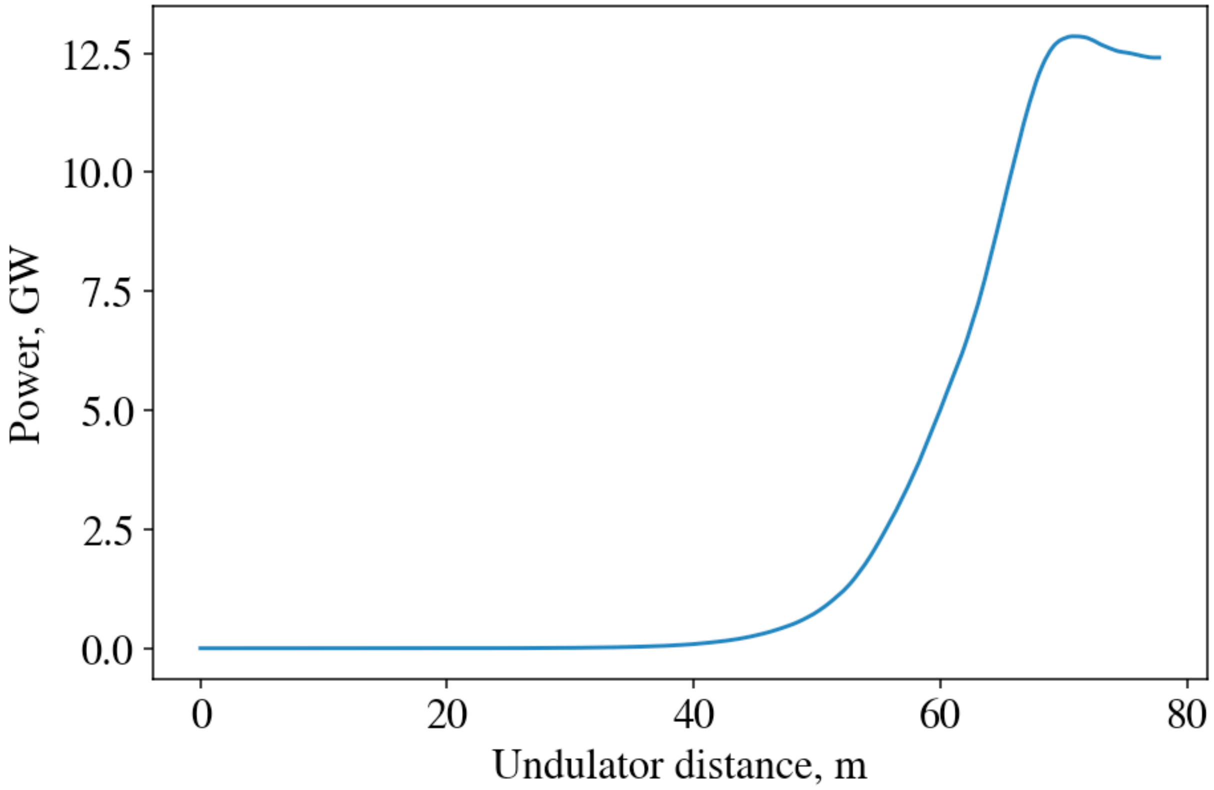


We look at the power growth along the undulator and see oscillations at the start;
We then estimate the Gain coefficient at the several points along the undulator;
The gain peaks at the resonant wavelength and not where the power picked!

LA-UR-21-20607

$\bar{\beta}$-function matching





$\bar{\beta}$-function matched solution produces a round beam in the undulator but not necessary the highest power!
Keeping a round beam may allow a smaller undulator gap and a stronger undulator parameter;
Our solution saturates at a longer undulator distance than Ming Xie predicted.

LA-UR-21-20607

Gain guiding of the optical mode