

**MELODY/MATLAB  
OBJECT-ORIENTED MODEL  
OF GRAVITATIONAL-WAVE INTERFEROMETERS  
USING MATLAB**

Raymond G. Beausoleil

*Stanford University/Hewlett-Packard Laboratories  
13837 175th Pl. NE, Redmond, WA 98052-2180*

beausol@hp1.hp.com

March 2002

LIGO-G020165-00-Z

# MELODY/MATLAB OVERVIEW

---

- Goals and features
- Propagation model
- Object-level features
  - Interferometer configurations
  - Mirror physics: thermal loading, position, orientation
  - Four-stage resonator length pseudolocking
- Script-level features
  - Modulation schemes
  - Mirror parameters: thermal, position, orientation
  - Full interactive MATLAB functionality
- Milestones

# ACKNOWLEDGMENTS

---

- *Stanford*

- Eric Gustafson
- Marty Fejer

- *Caltech*

- Erika D'Ambrosio
- Bill Kells
- Jordan Camp

- *MIT*

- Ryan Lawrence
- Peter Frischel

- *Glasgow*

- Ken Strain

- *UCF*

- Guido Mueller

- *LIGO*

- Daniel Sigg

# MELODY/MATLAB GOALS

---

- Provide an easily usable, flexible multiplatform framework for LIGO I/II/III calculations and simulations
- Allow users to write scripts to drive simulations tailored to their needs (post-processing, graphics, numerical analysis)
- Easily include physical effects in mirrors: aperture diffraction, curvature mismatch, thermal lensing, thermoelastic surface deformation
- Allow translation to a lower-level language for performance
- Provide a simple interface to industry-standard software for modeling control systems (SIMULINK)

# MELODY/MATLAB FEATURES

---

- MATLAB classes for fields, mirrors, interferometers, and detectors; driven by user-written scripts → self-consistent solutions
- Prebuilt LIGO I/II configurations
  - Power, signal, and dual recycling
  - Arbitrary modulation schemes
  - Resonator length pseudolocking for self-contained simulations
- Mirror physics
  - Aperture diffraction
  - Mirror surface/laser wavefront curvature mismatch
  - Thermal lensing due to bulk and coating absorption (TEM<sub>00</sub>)
  - *Thermoelastic surface deformation (reflection, transmission)*

# NEW MELODY/MATLAB FEATURES

---

- Numerical FEMLAB model of thermal wavefront distortion in tilted flat optics (Ryan Lawrence)
  - Arbitrary angle of incidence
  - substrate, HR and AR coating absorption
  - Thermal focusing *and* thermoelastic surface deformation
  - Melody integration and testing underway
- Quadratic thermoelastic surface curvature reporting at runtime
- New Gingin class
- Substrate OPD inhomogeneities (power-independent phase maps)

# MELODY/MATLAB LIMITATIONS

---

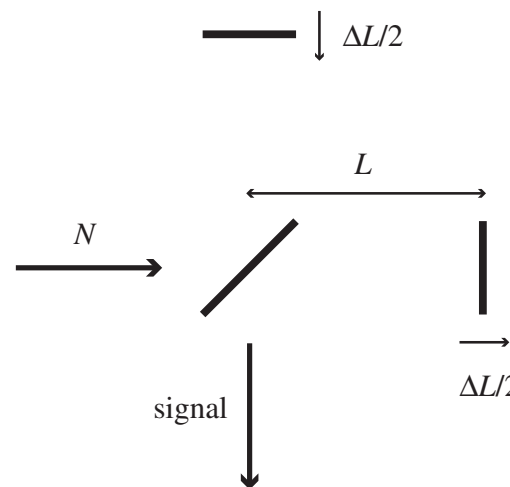
- Models thermal loading due to TEM<sub>00</sub> absorption only, summed over all frequency components
- Transient thermal loading not yet implemented — calculations complete, but will require nontrivial architectural changes

# GRAVITATIONAL WAVE DETECTION

---

$$\Delta\Phi \approx \frac{\Delta L}{L} \frac{L}{\lambda} \equiv h \frac{L}{\lambda} > \frac{1}{\sqrt{N}}$$

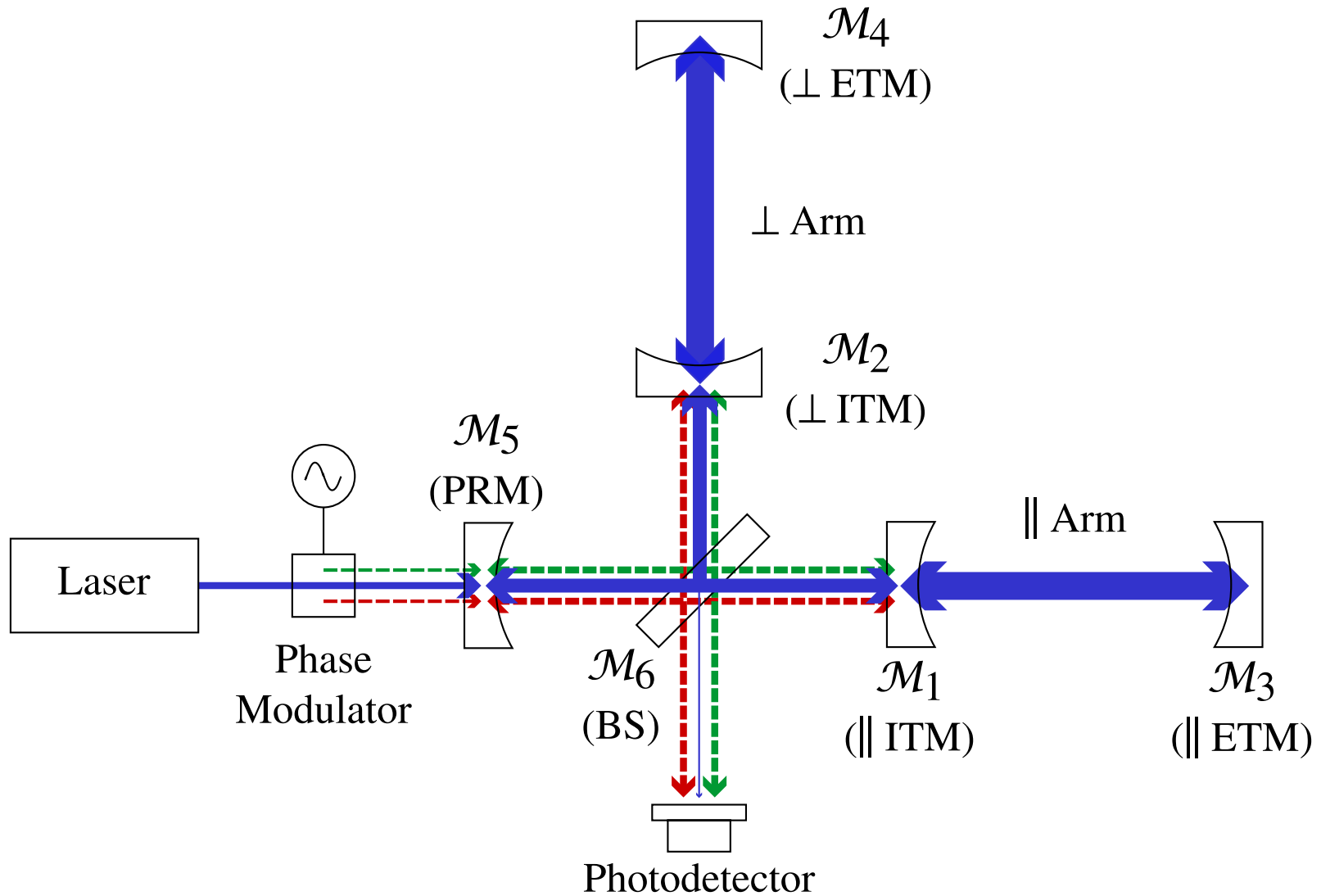
$$h_{\min} \approx \frac{\lambda}{L} \frac{1}{\sqrt{N}}$$



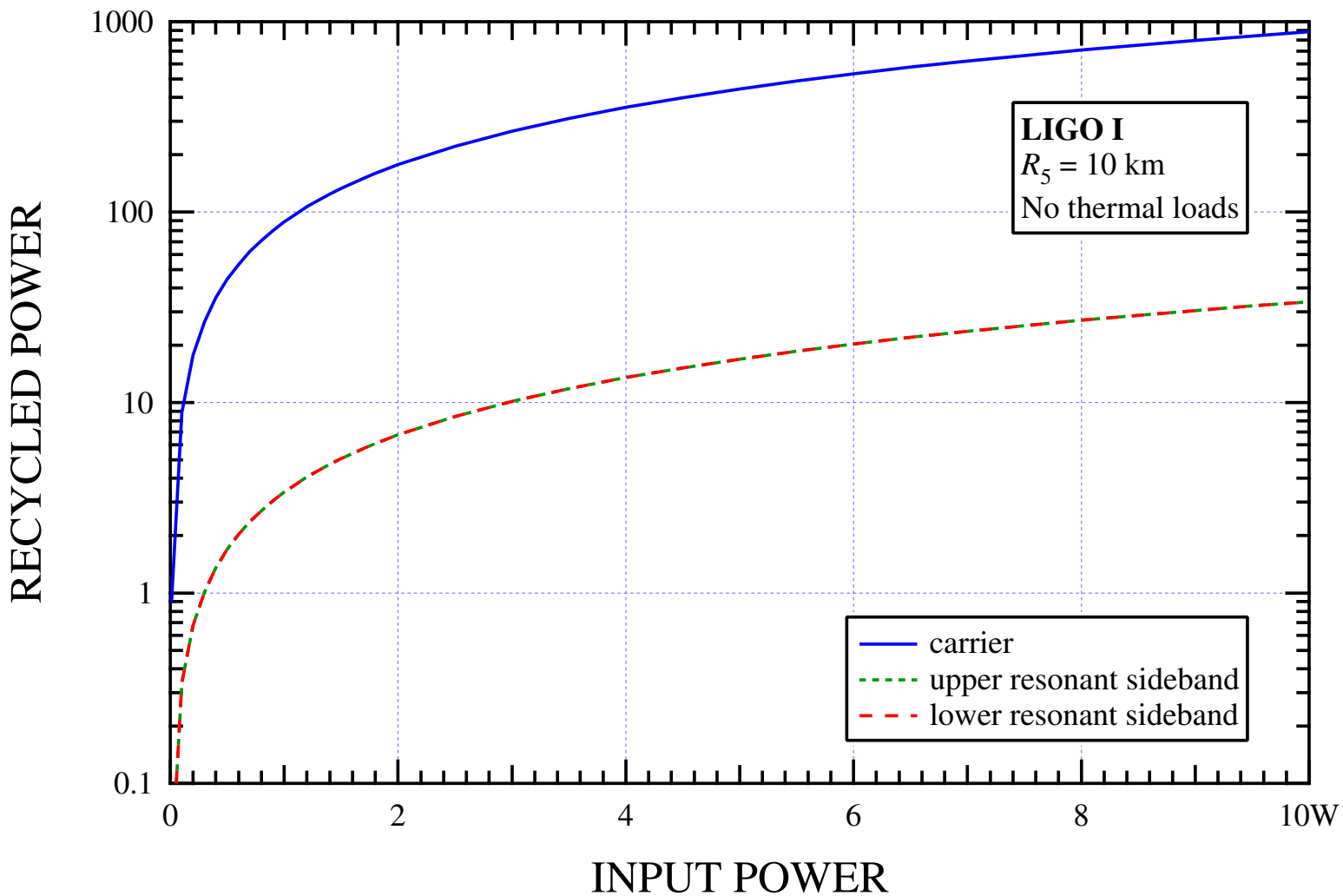
- Use Fabry-Perot interferometers:  $L \rightarrow B L$ 
  - Improved sensitivity
  - Longer storage time  $\rightarrow$  lower signal frequencies
- Dark fringe operation  $\rightarrow$  use power recycling
  - Improved sensitivity



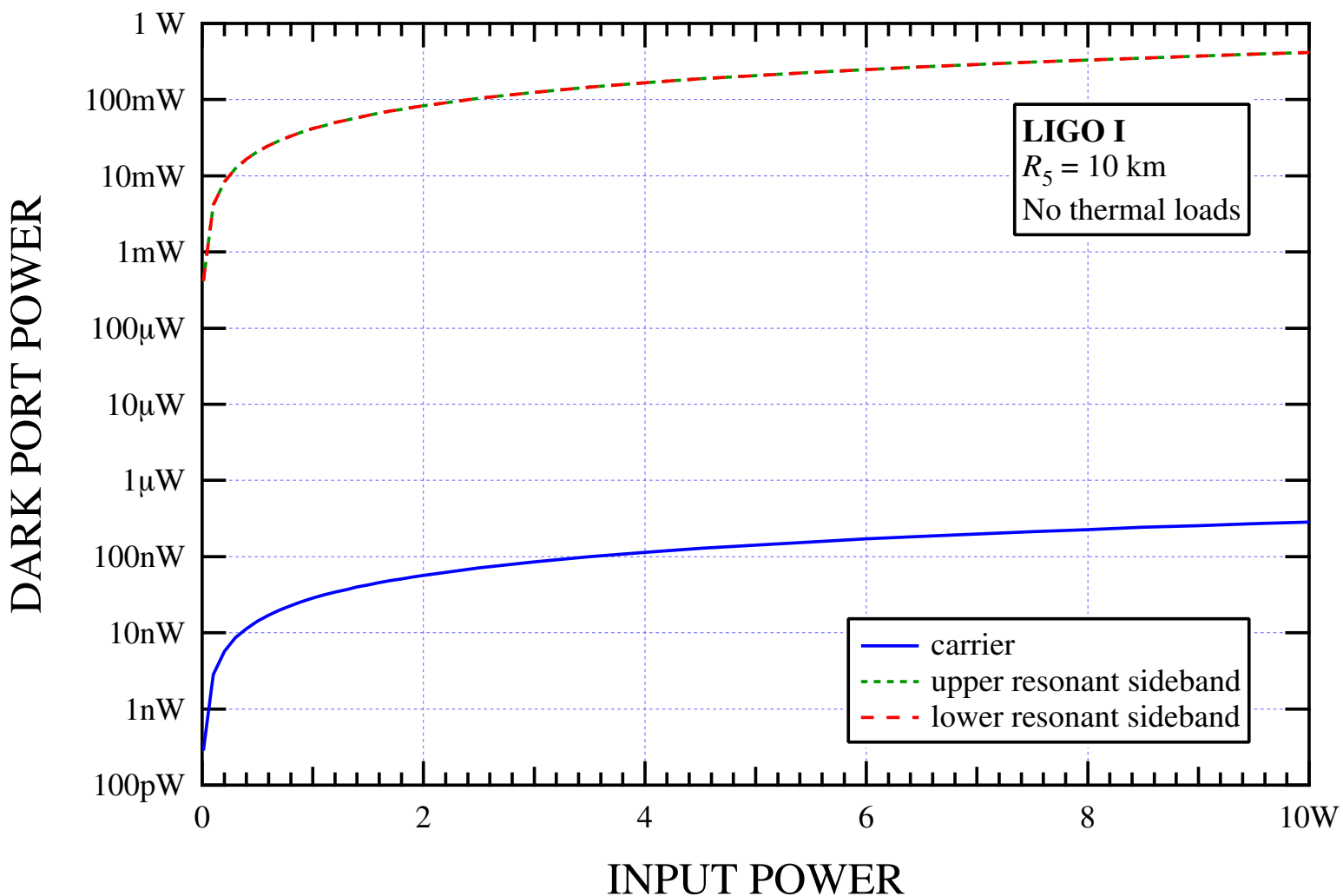
# LIGO IFO CONFIGURATION



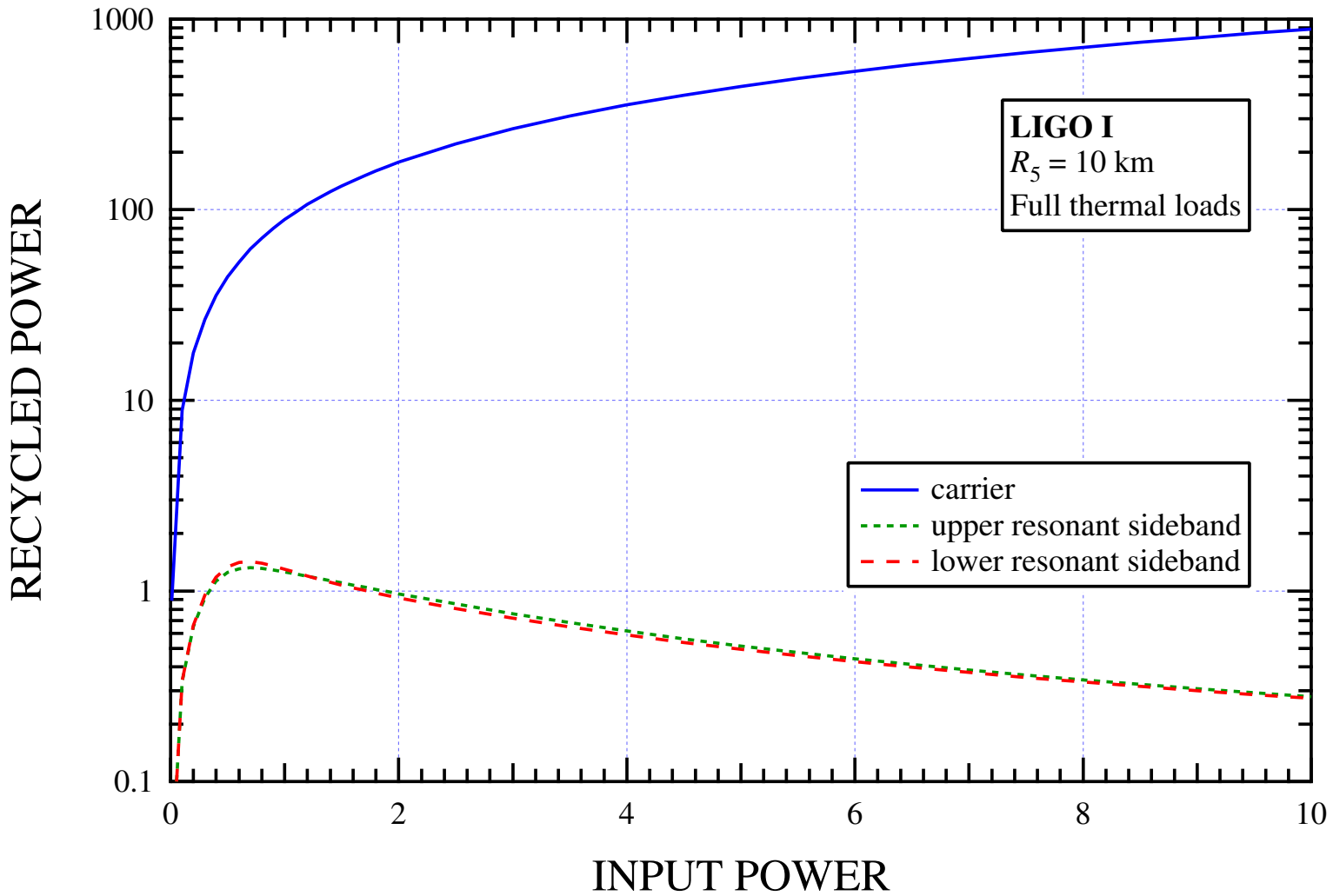
# LIGO I: COLD, $R_5 = 10$ km, $E_0 \equiv \text{TEM}_{00}$



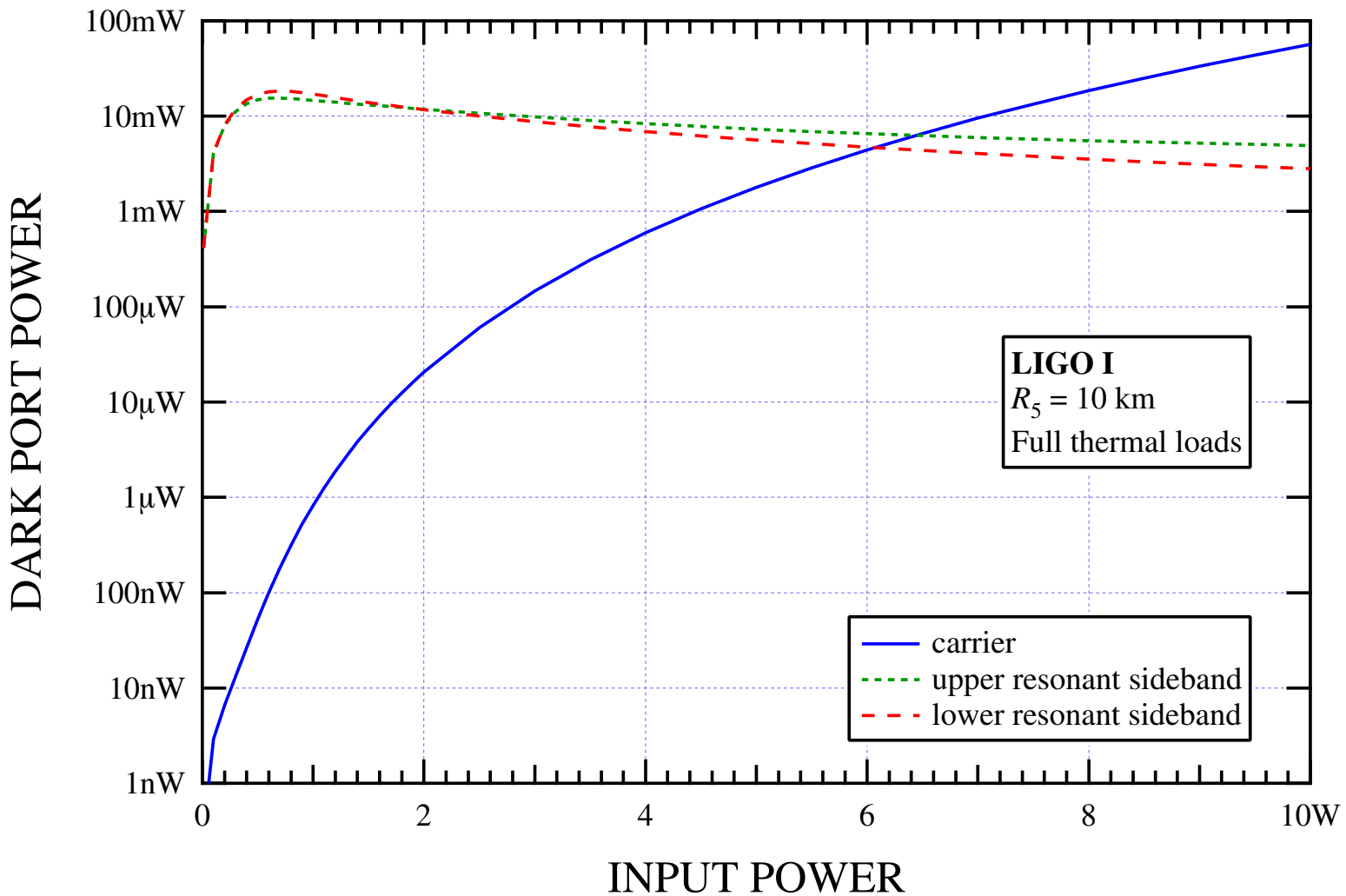
# LIGO I: COLD, $R_5 = 10$ km, $E_0 \equiv \text{TEM}_{00}$



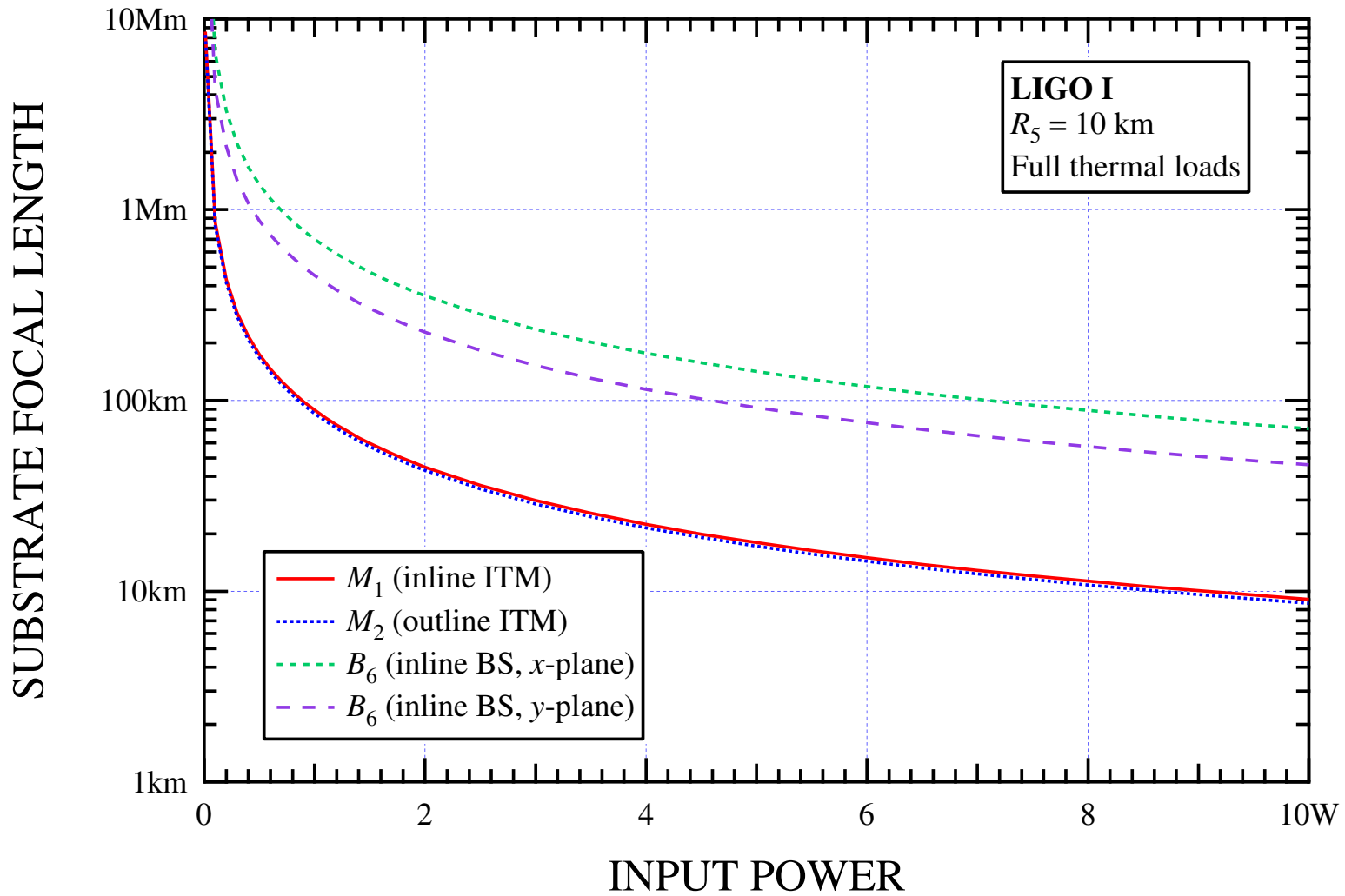
# LIGO I: HOT, $R_5 = 10$ km, $E_0 \equiv \text{TEM}_{00}$



# LIGO I: HOT, $R_5 = 10$ km, $E_0 \equiv \text{TEM}_{00}$



# LIGO I: TFL, $R_5 = 10$ km, $E_0 \equiv \text{TEM}_{00}$

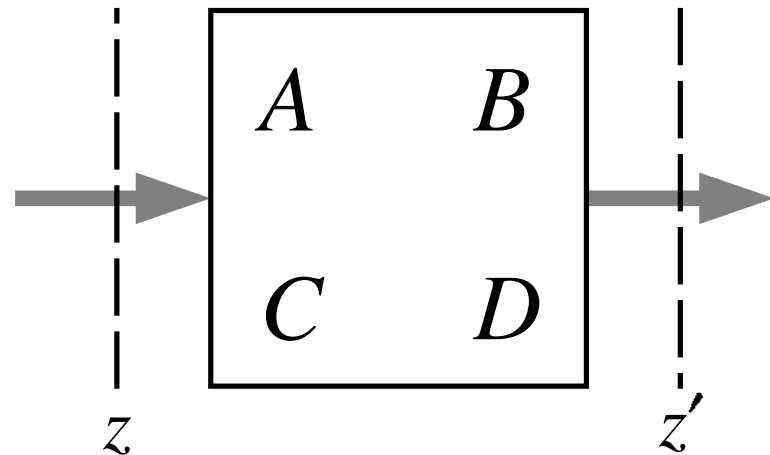


# FORWARD PROPAGATION: HUYGENS-FRESNEL INTEGRAL

---

$$\mathbf{E}(\mathbf{r}, t) \equiv \text{Re} \left\{ \epsilon E(\mathbf{r}) e^{i(kz - \omega t)} \right\}$$

$$\nabla_{\perp}^2 E(\mathbf{r}) + i2k \frac{\partial}{\partial z} E(\mathbf{r}) = 0$$



Huygens Integral:

$$E(x, y, z) = \int_{\mathcal{A}_1} dx' dy' K(x, y; x', y') E(x', y', z') \equiv \hat{K}[E(x', y', z')]$$

Fresnel Approximation:

$$K(x, y; x', y') =$$

$$\frac{1}{i\lambda B} \exp \left\{ i \frac{\pi}{\lambda B} \left[ A(x'^2 + y'^2) - 2(x'x + y'y) + D(x^2 + y^2) \right] \right\}$$

# UNPERTURBED BASIS FUNCTIONS

---

Forward and backward unperturbed basis functions:

$$y_{mn} u_{mn}(x, y, 0) = \int_{\mathcal{A}_1} dx' dy' K_0(x, y; x', y') u_{mn}(x', y', 0)$$

$$y_{mn}^\dagger u_{mn}^\dagger(x, y, 0) = \int_{\mathcal{A}_1} dx' dy' K_0^\dagger(x, y; x', y') u_{mn}^\dagger(x', y', 0)$$

Biorthogonality relation (Siegman), satisfied discretely:

$$\int_{\mathcal{A}_1} dx dy u_{mn}^\dagger(x, y, z) u_{m'n'}(x, y, z) = \delta_{mm'} \delta_{nn'}$$

Expand intracavity field:

$$E(x, y, z, t) = \sum_{mn} E_{mn}(t) u_{mn}(x, y, z)$$



# PROPAGATOR MATRIX ELEMENTS

---

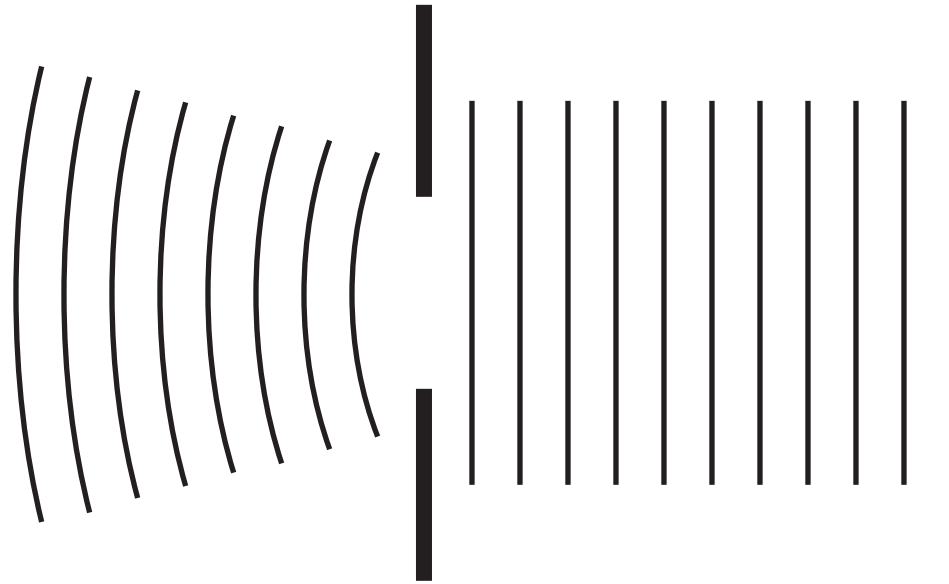
Calculate  $K_{mn;m'n'}(t)$  as the matrix element of the fully perturbed forward propagator (from reference plane  $\mathcal{A}_1$  to reference plane  $\mathcal{A}_2$ ) in the unperturbed basis:

$$K_{mn;m'n'}(t) = \int_{\mathcal{A}_2} dx dy \int_{\mathcal{A}_1} dx' dy' \\ \times u_{mn}^\dagger(x, y) K(x, y; x', y'; t) u_{m'n'}(x', y')$$

We compute  $K_{mn;m'n'}(t)$  for each propagation region in the extended unperturbed basis of the interferometer; then construct a representation of the perturbed interferometer using matrix multiplication.

# APERTURE DIFFRACTION

---



$$\begin{aligned} A_{mn,m'n'} &= \iint_{\mathcal{A}} dx dy u_{mn}^*(x, y) u_{m'n'}(x, y) \\ &\equiv \delta_{m,m'} \delta_{n,n'} - e^{-\alpha} I_{mn,m'n'}(\alpha), \end{aligned}$$

where  $\alpha \equiv 2(a/w)^2$ , and nonzero elements of  $I(\alpha)$  satisfy

$$I_{mn,m'n'}(\alpha) \approx O \left[ \alpha^{\frac{1}{2}(m+n+m'+n')} \right]$$

# CURVATURE MISMATCH

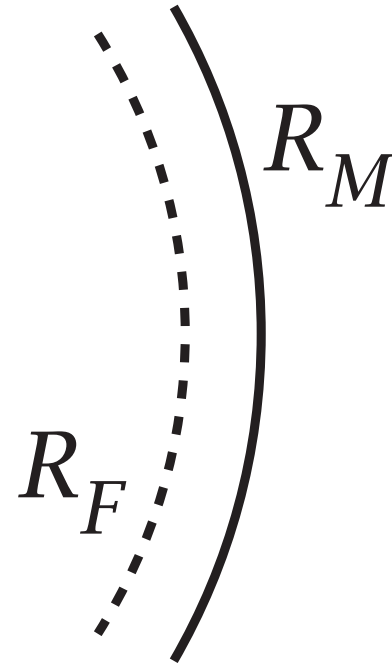
---

Unitary approximation:

$$C \cong \exp(i \gamma c),$$

where

$$\gamma \equiv \frac{\pi w^2}{\lambda} \left( \frac{1}{R_F} - \frac{1}{R_M} \right),$$



and, in the Hermite-Gauss basis,

$$\begin{aligned} c_{mn,m'n'} &\equiv \frac{2}{w^2} \iint_{-\infty}^{\infty} dx dy |u_{mn}(x, y) u_{m'n'}(x, y)| (x^2 + y^2) \\ &= X_{m,m'}^2 \delta_{n,n'} + \delta_{m,m'} X_{n,n'}^2 \end{aligned}$$

# THERMAL LENSING

---

- Hello-Vinet model of substrate thermal lensing due to both substrate and coating absorption
- H-V bulk absorption result agrees with approximations:

- Infinite half-space approximation:

$$T(r) - T(0) = -\frac{\alpha_P P}{4\pi k_T} \left[ \gamma + \ln \left( \frac{2r^2}{w^2} \right) + E_1 \left( \frac{2r^2}{w^2} \right) \right]$$

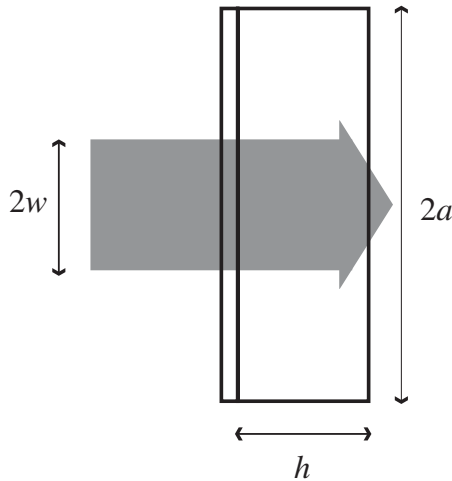
- Near  $r = 0$ , thin lens approximation:

$$f = \frac{\pi w^2}{\alpha_P h P} \frac{\kappa_T}{dn/dT}$$

- Numerical implementation of astigmatic thermal loading in beam-splitter almost complete (Hermite-Gauss basis)

# HELLO-VINET THERMAL LENS MODEL

---



Reference: P. Hello and J.-Y. Vinet,  
J. Phys. France 51, 1267 (1990)

Coating absorption:

$$T_c(r, z) = \frac{P_c}{k_T a} \sum_{k=0}^{\infty} a^2 p_k \left[ A_k \cosh \left( \zeta_k \frac{z}{a} \right) + B_k \sinh \left( \zeta_k \frac{z}{a} \right) \right] J_0 \left( \zeta_k \frac{r}{a} \right)$$

Substrate absorption:

$$T_s(r, z) = \frac{P_s}{k_T h} \sum_{k=0}^{\infty} \frac{a^2 p_k}{\zeta_k^2} \left[ 1 - 2\tau A_k \cosh \left( \zeta_k \frac{z}{a} \right) \right] J_0 \left( \zeta_k \frac{r}{a} \right)$$

# HELLO-VINET THERMAL CONSTANTS

---

$\zeta_k$ : Roots of the equation

$$\zeta J_1(\zeta) - \tau J_0(\zeta) = 0$$

Since  $\tau \equiv 4\epsilon T^3 a / k_T = 0.27734$  for fused silica at room temperature,

$$\zeta_k \cong (k + 1/4) \pi, \quad k \in \{0, 1, 2, \dots\}$$

$p_k$ : Normalized expansion coefficients

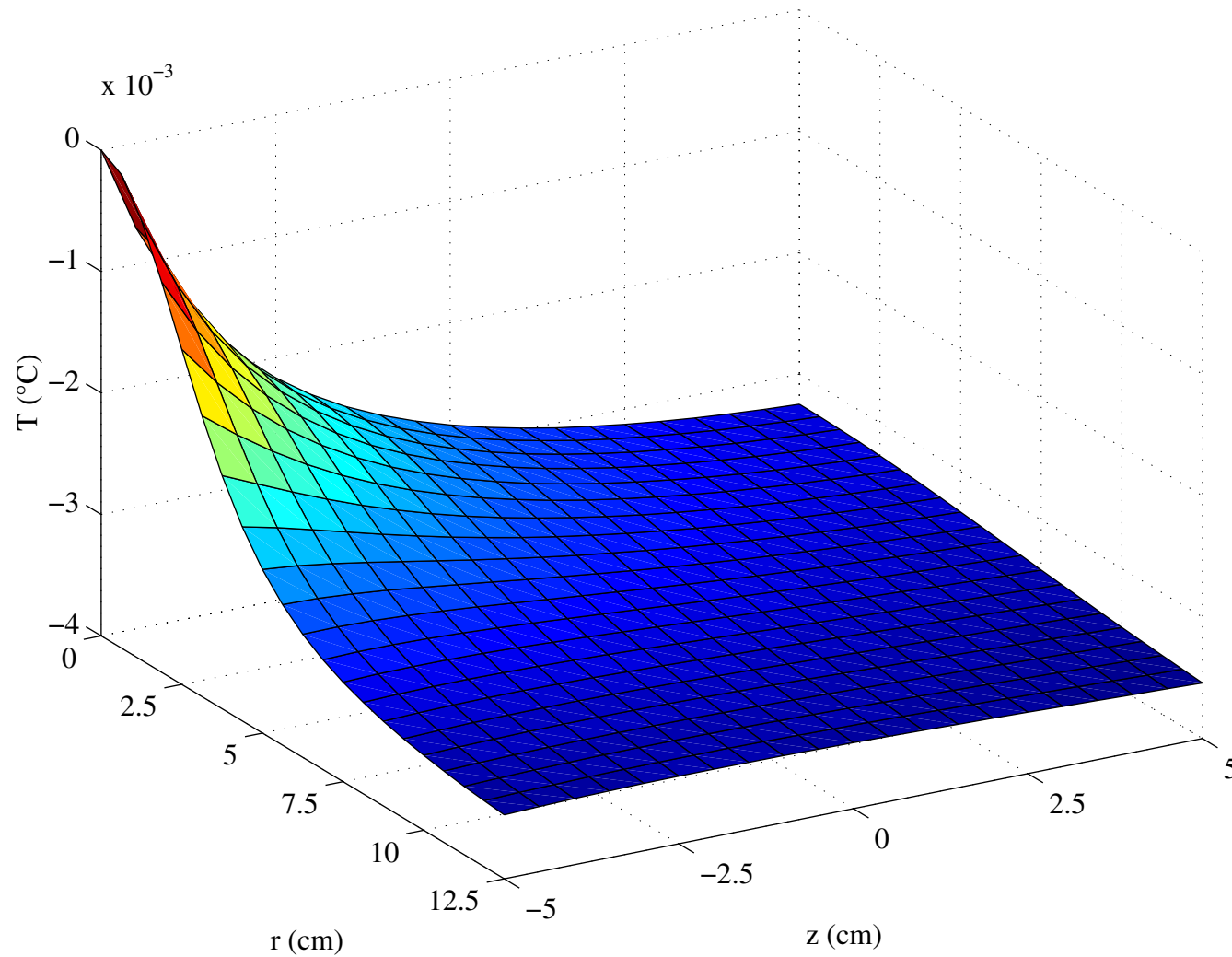
$$u_{00}^2 = \sum_{k=0}^{\infty} p_k J_0\left(\zeta_k \frac{r}{a}\right)$$

Since  $(w/a)^2 \ll 1$ ,

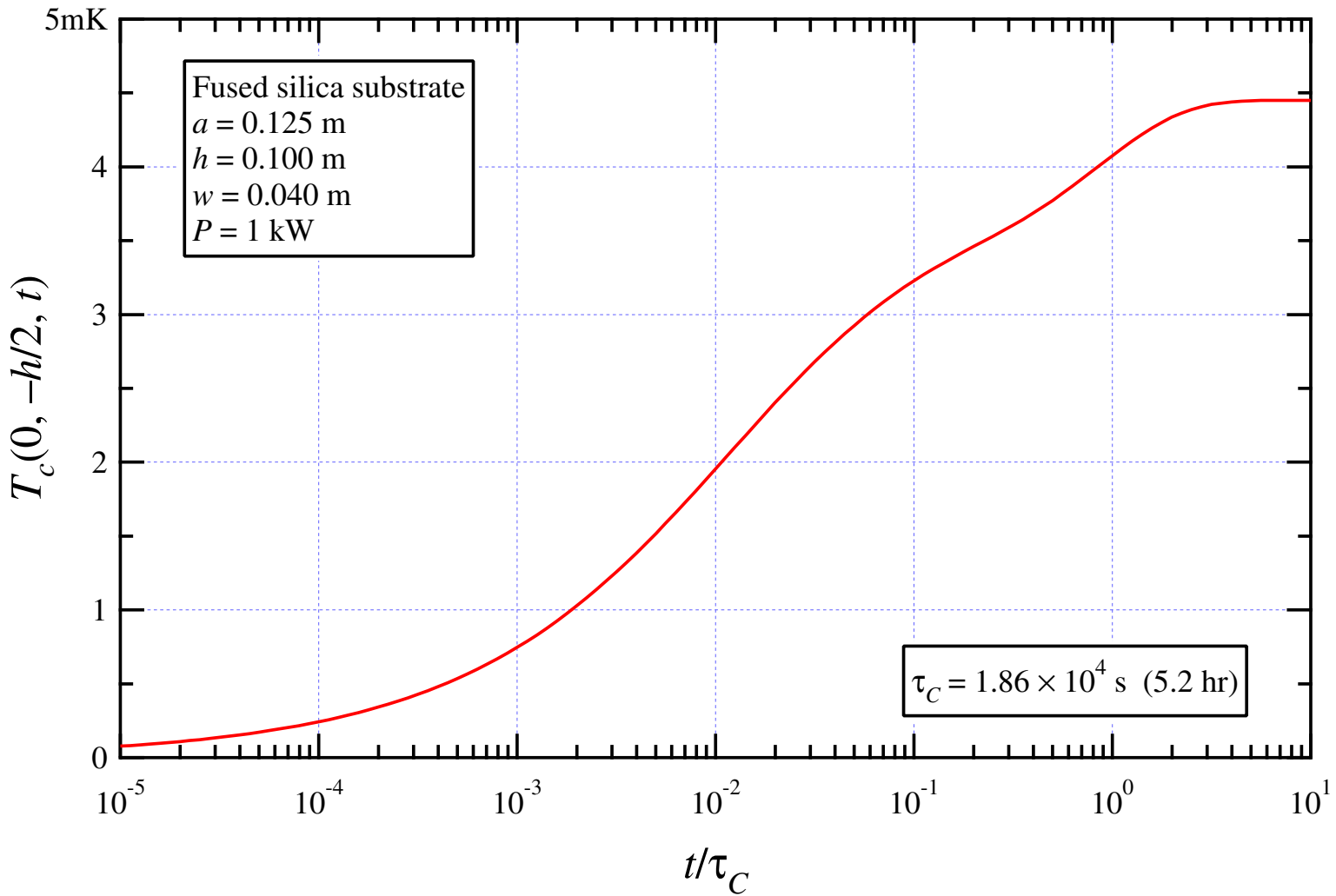
$$p_k \cong \frac{P}{\pi a^2} \frac{\zeta_k^2}{(\zeta_k^2 + \tau^2) J_0^2(\zeta_k)} e^{-(\zeta_k w/a)^2/8}$$

# $T_c(r, z)$ FROM COATING ABSORPTION

---

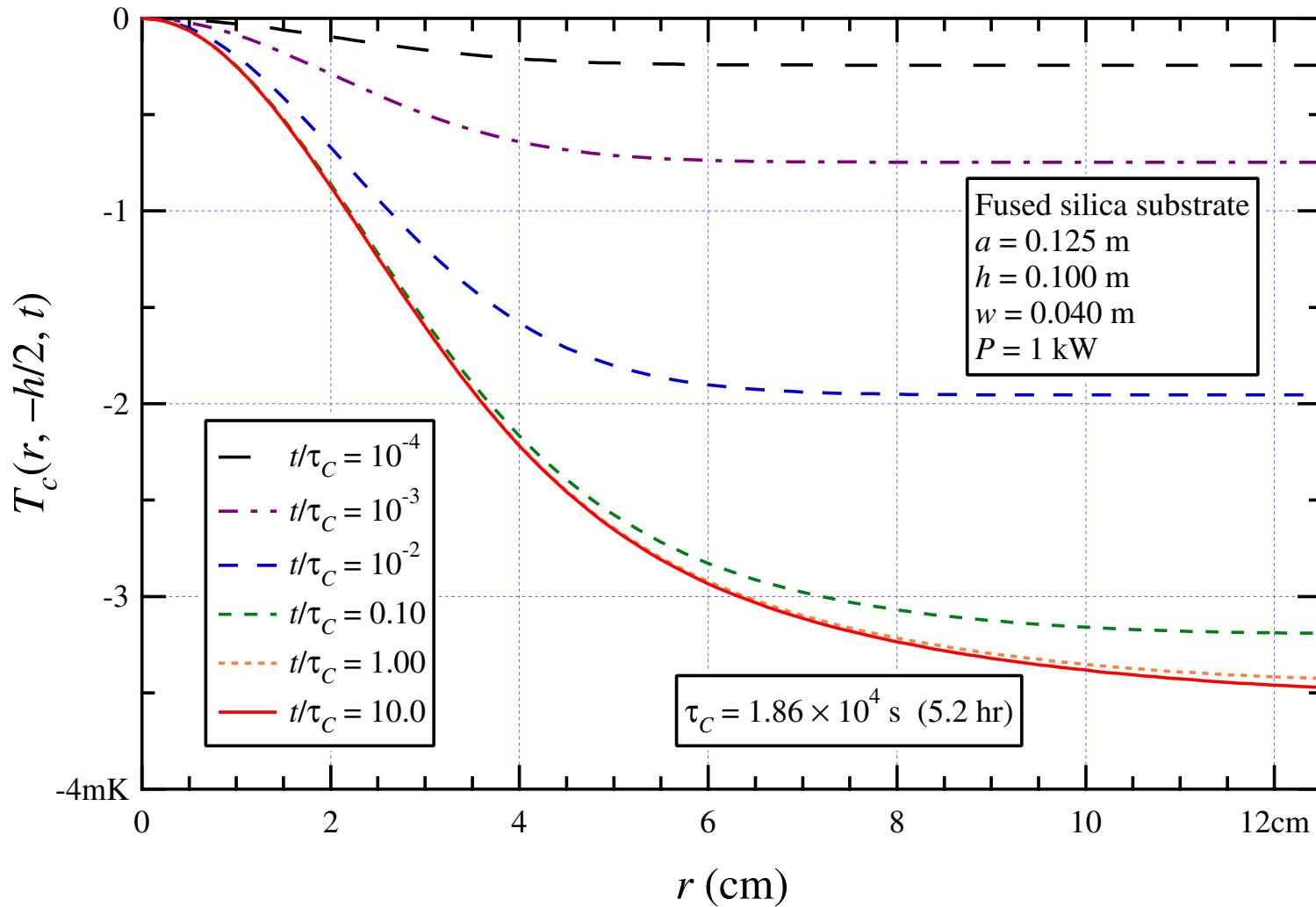


# $T_c(0, -h/2, t)$ FROM COATING ABSORPTION

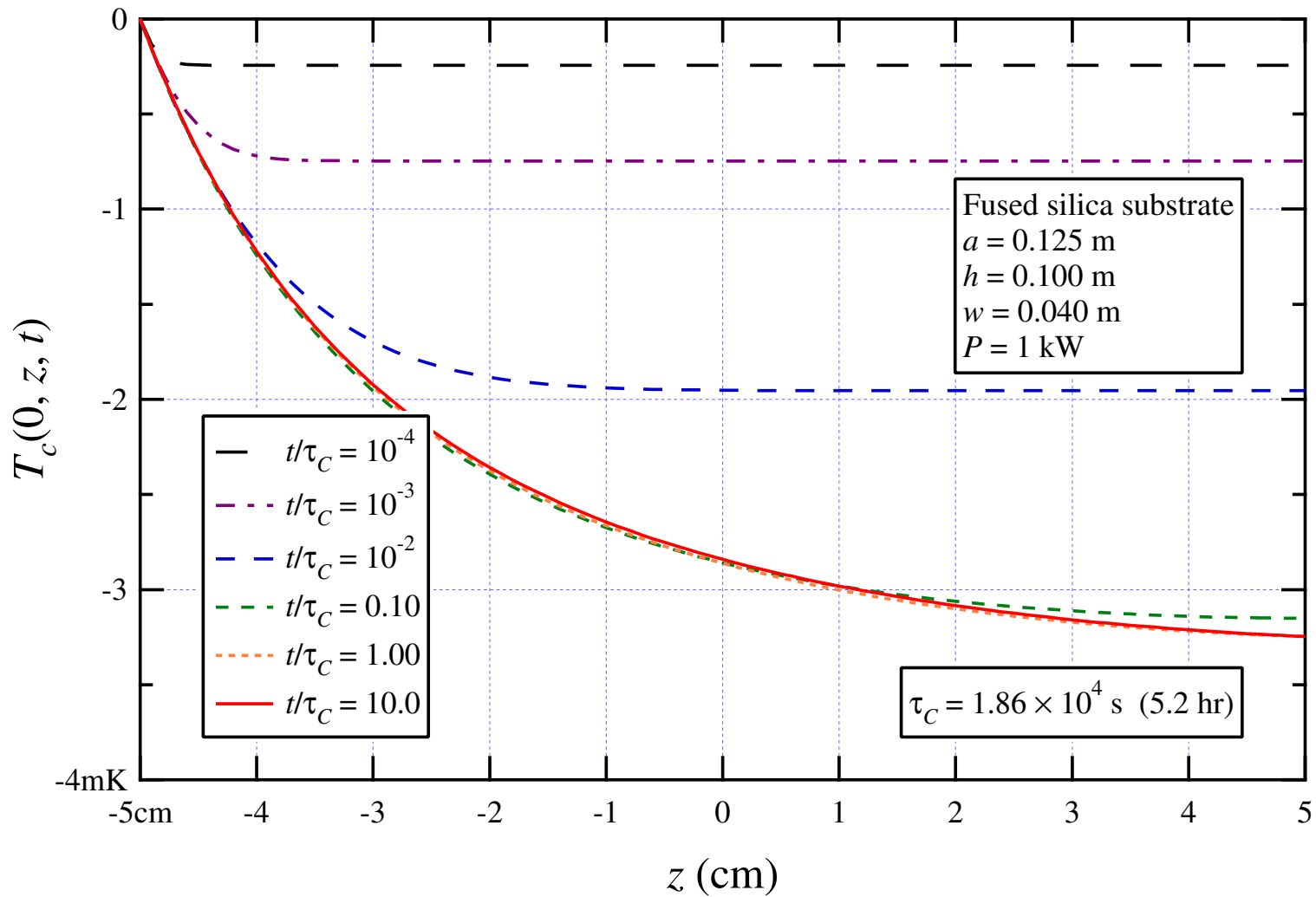




# $T_c(r, -h/2, t)$ FROM COATING ABSORPTION

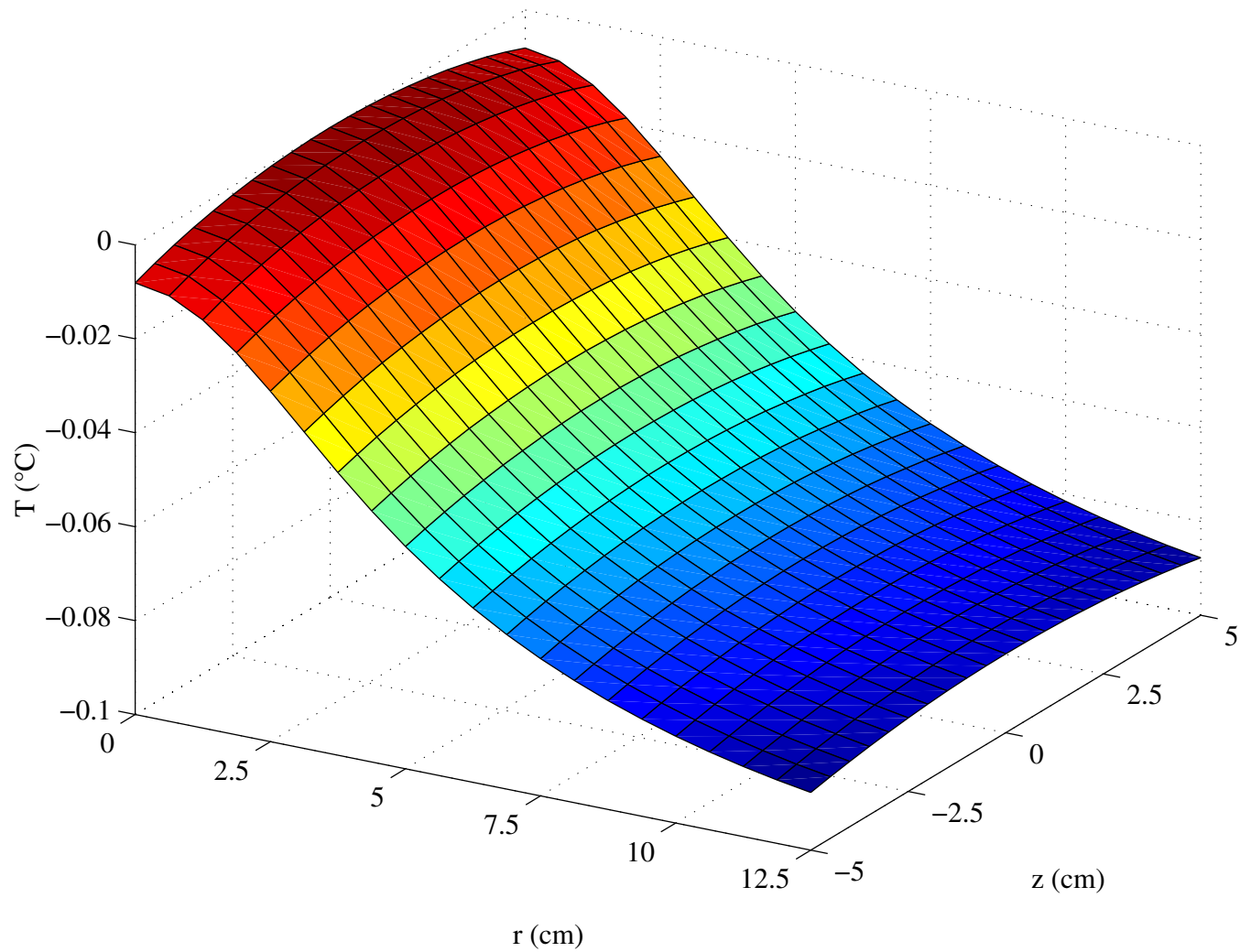


# $T_c(0, z, t)$ FROM COATING ABSORPTION

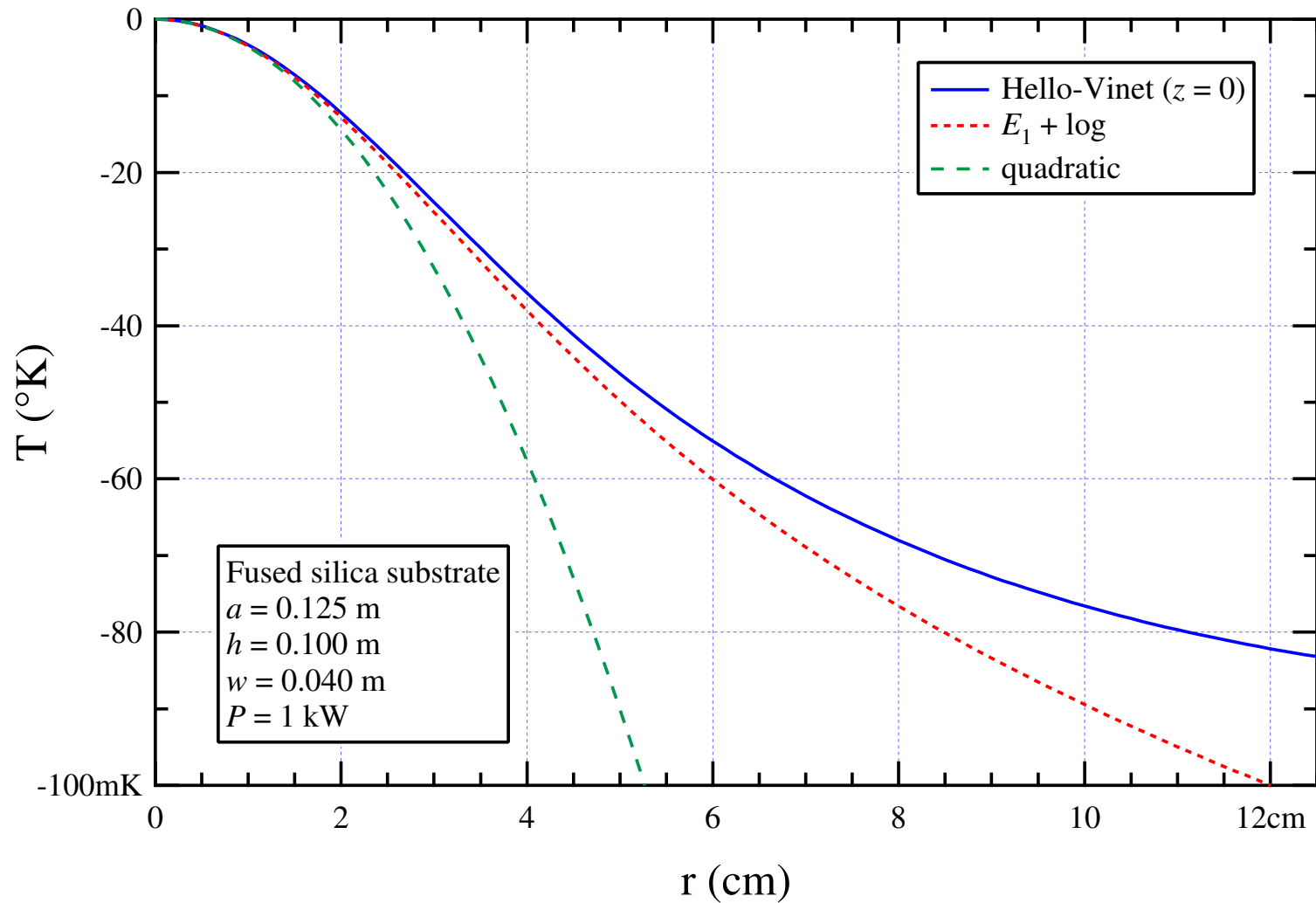


# $T_S(r, z)$ FROM SUBSTRATE ABSORPTION

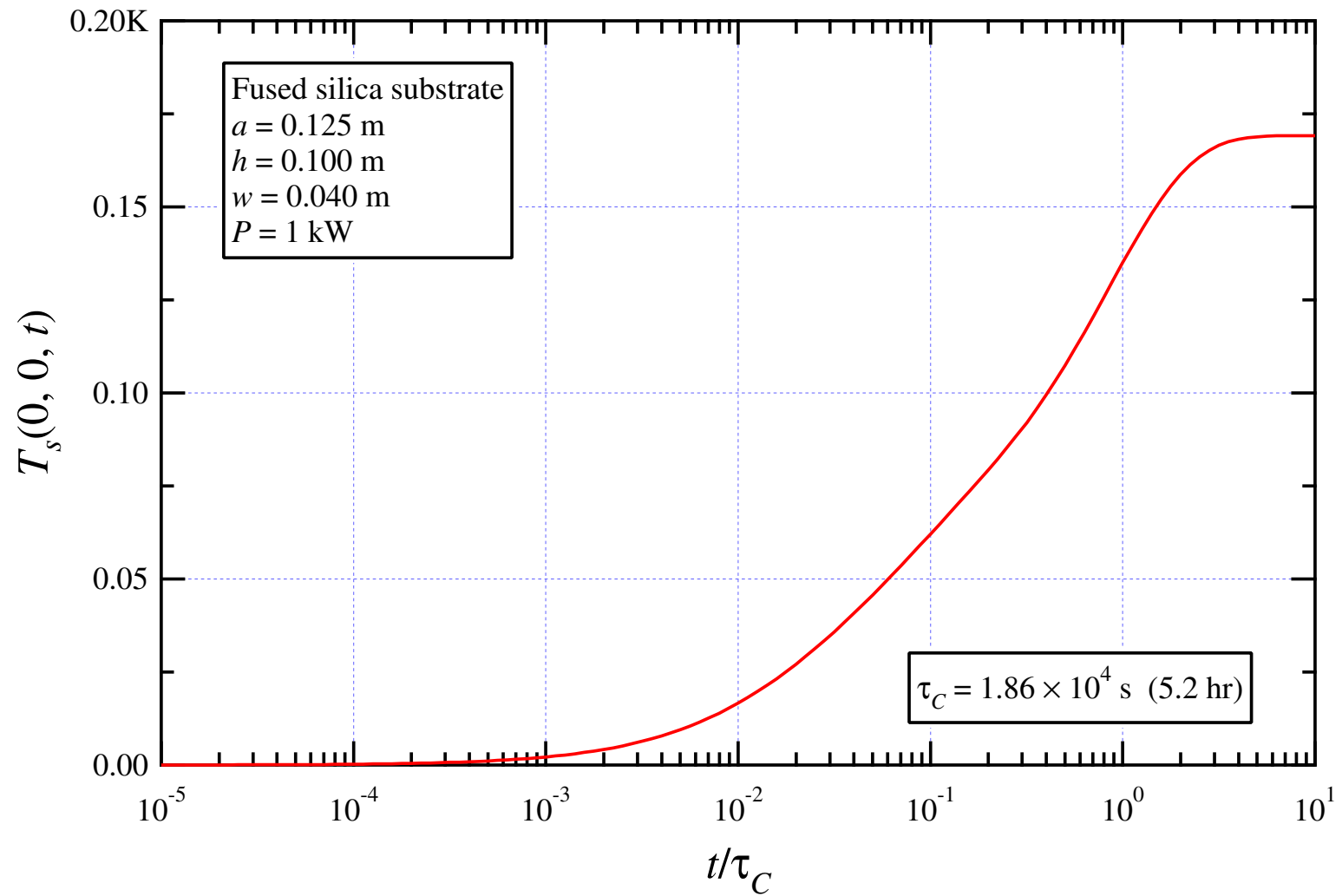
---



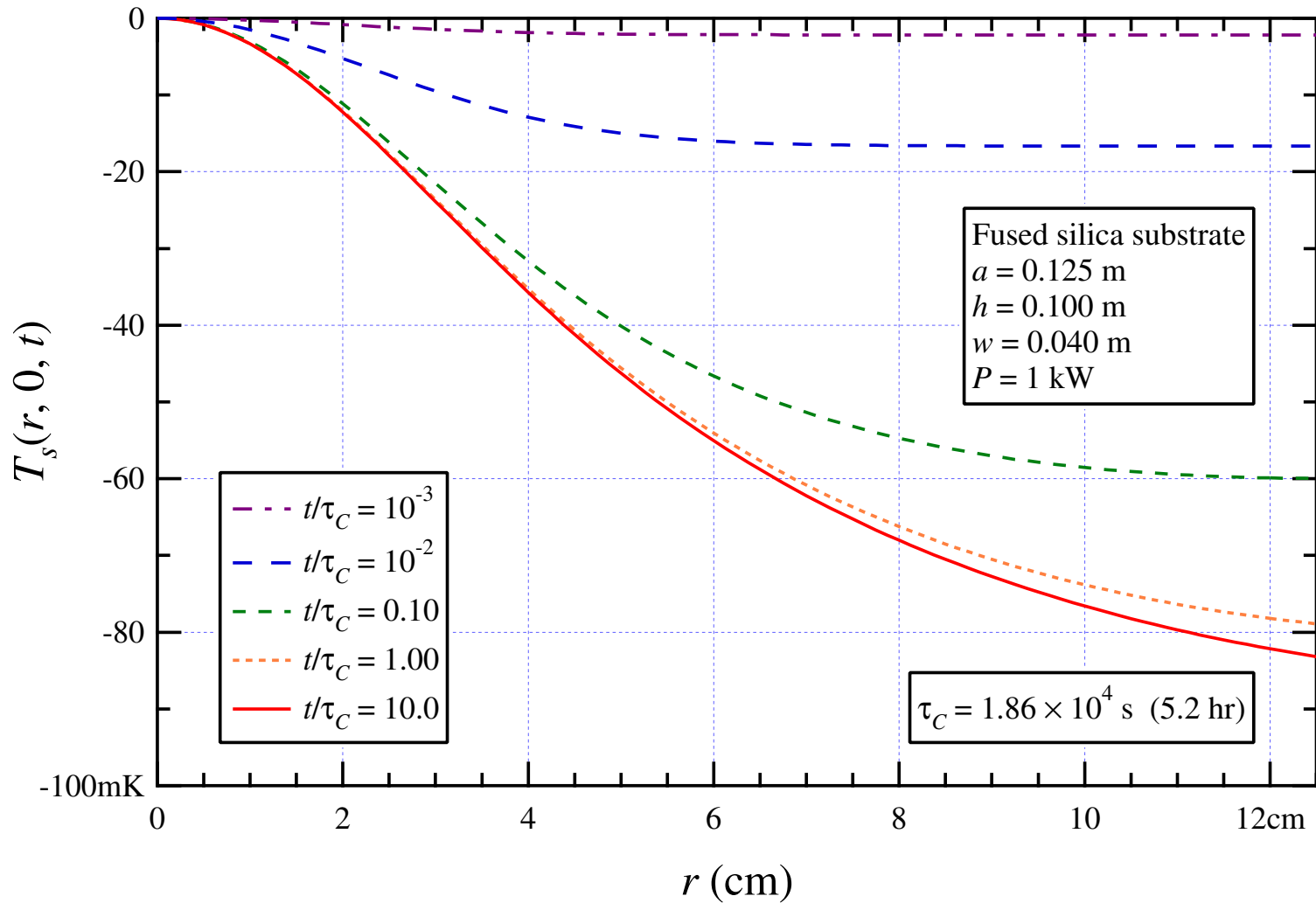
# APPROXIMATE $T_s(r, 0)$ (SUBSTRATE ABSORPTION)



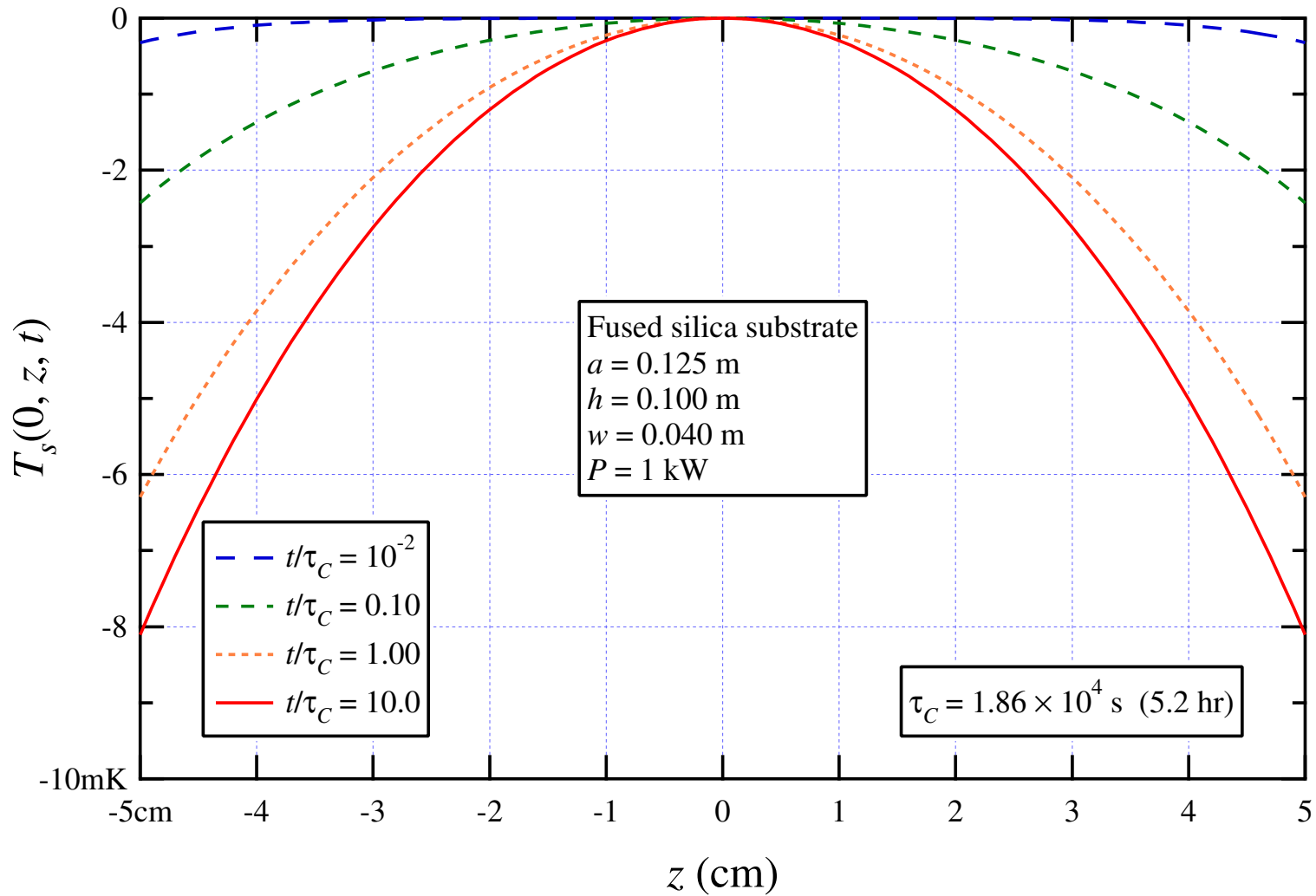
# $T_s(0, 0, t)$ FROM SUBSTRATE ABSORPTION



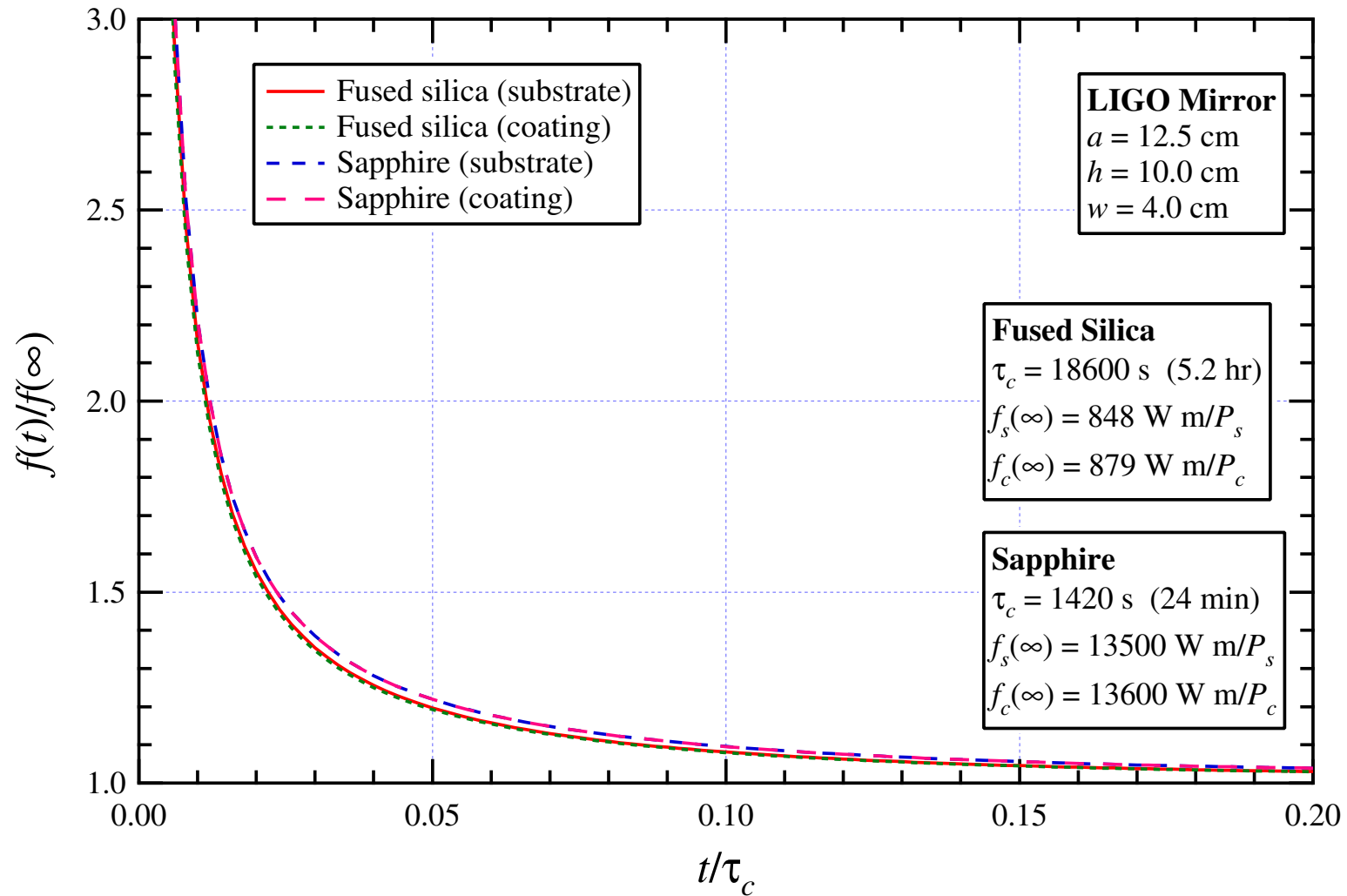
# $T_s(r, 0, t)$ FROM SUBSTRATE ABSORPTION



# $T_S(0, z, t)$ FROM SUBSTRATE ABSORPTION



# THERMAL FOCAL LENGTH





# THERMAL LENS OPERATOR

---

The propagation phase perturbation due to the substrate OPD is

$$\phi(r) = \frac{2\pi}{\lambda_0} \frac{dn}{dT} \int_{-h/2}^{h/2} dz T(r, z)$$

where  $T(r, z)$  is the *linear* sum of contributions from heating due to absorption in both coatings (HR and AR) and the substrate.

Matrix elements of the thermal lens operator (per unit power absorbed):

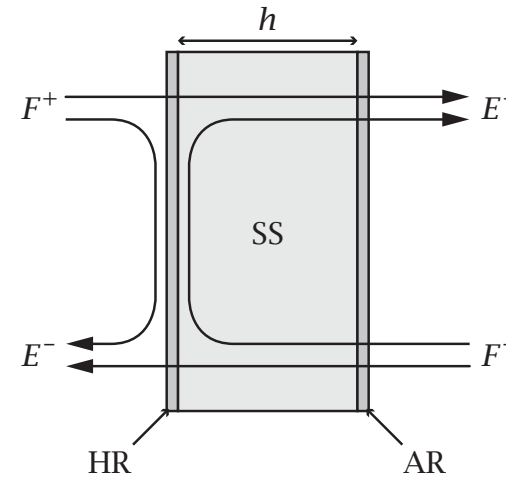
$$\Phi_{m'n';mn} = \iint_{-\infty}^{\infty} dx dy u_{m'n'}^\dagger(x, y) u_{mn}(x, y) \phi(r)$$

Since  $\phi(r) \propto r^2$ , TEM<sub>00</sub> is coupled to both TEM<sub>20</sub> and TEM<sub>02</sub>.

# FINAL THERMAL LENS MATRICES

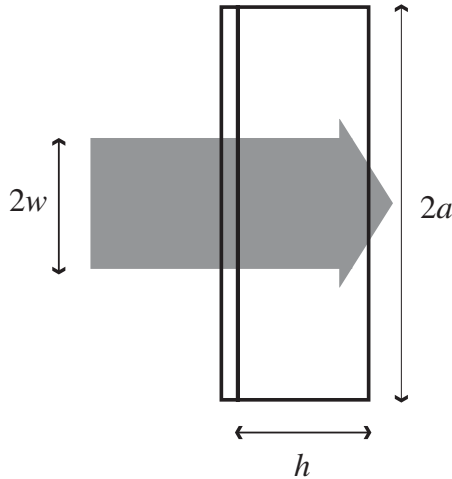
Unitary approximation:

$$S = \exp \{i [P_s \Phi_s + (P_h + P_a) \Phi_c]\}$$



Region	Absorbed Power
SS	$P_s = \alpha_s h \frac{1}{2} \sum_q ( E_{00q}^+ ^2 +  F_{00q}^- ^2)$
HR	$P_h = a_{hr} \frac{1}{2} \sum_q ( F_{00q}^+ ^2 +  F_{00q}^- ^2)$
AR	$P_a = a_{ar} \frac{1}{2} \sum_q ( E_{00q}^+ ^2 +  F_{00q}^- ^2)$

# HELLO-VINET THERMOELASTIC SURFACE DEFORMATION



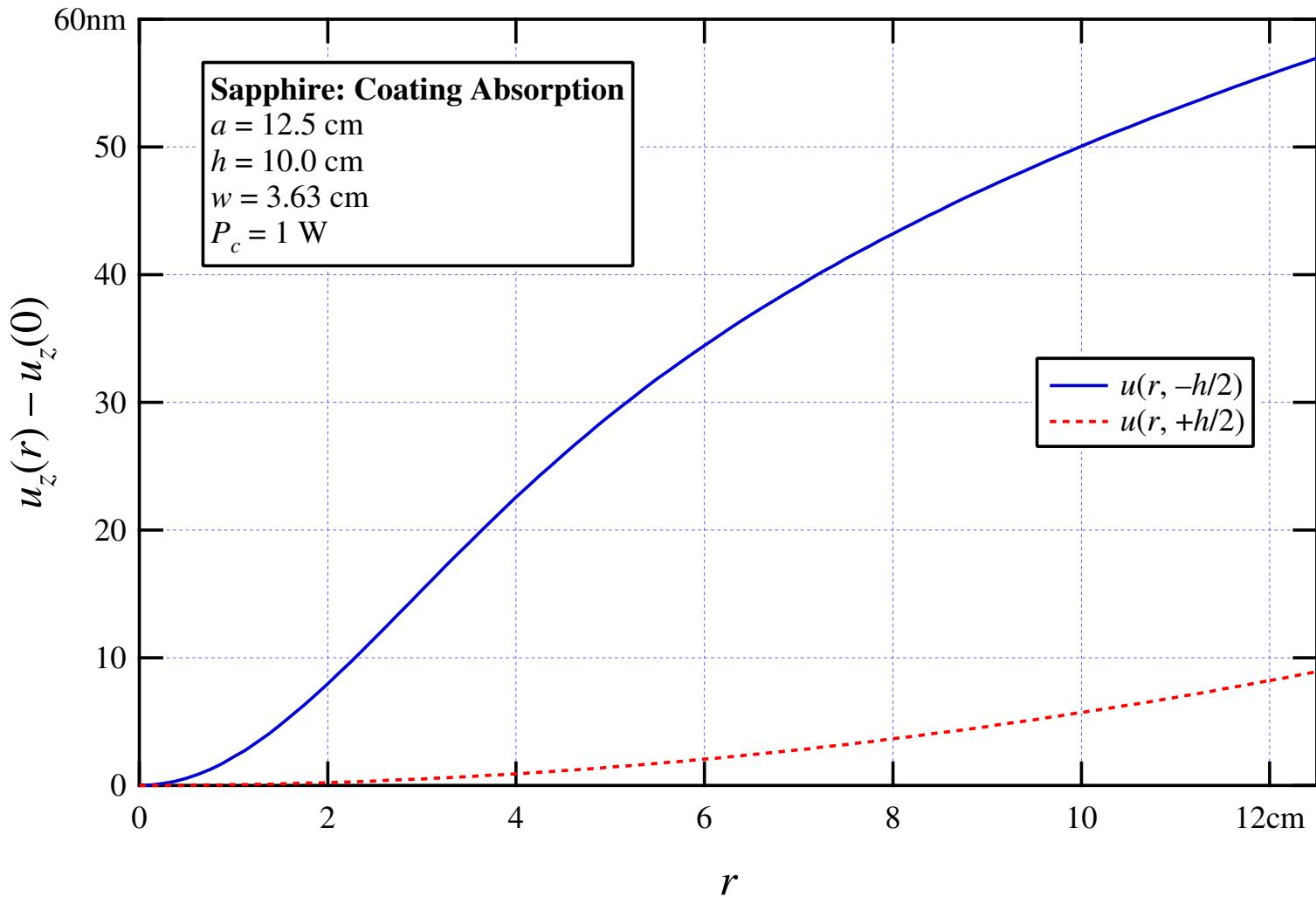
Reference: P. Hello and J.-Y. Vinet,  
J. Phys. France 51, 2243 (1990)

$$\gamma_k \equiv \zeta_k h / 2a$$

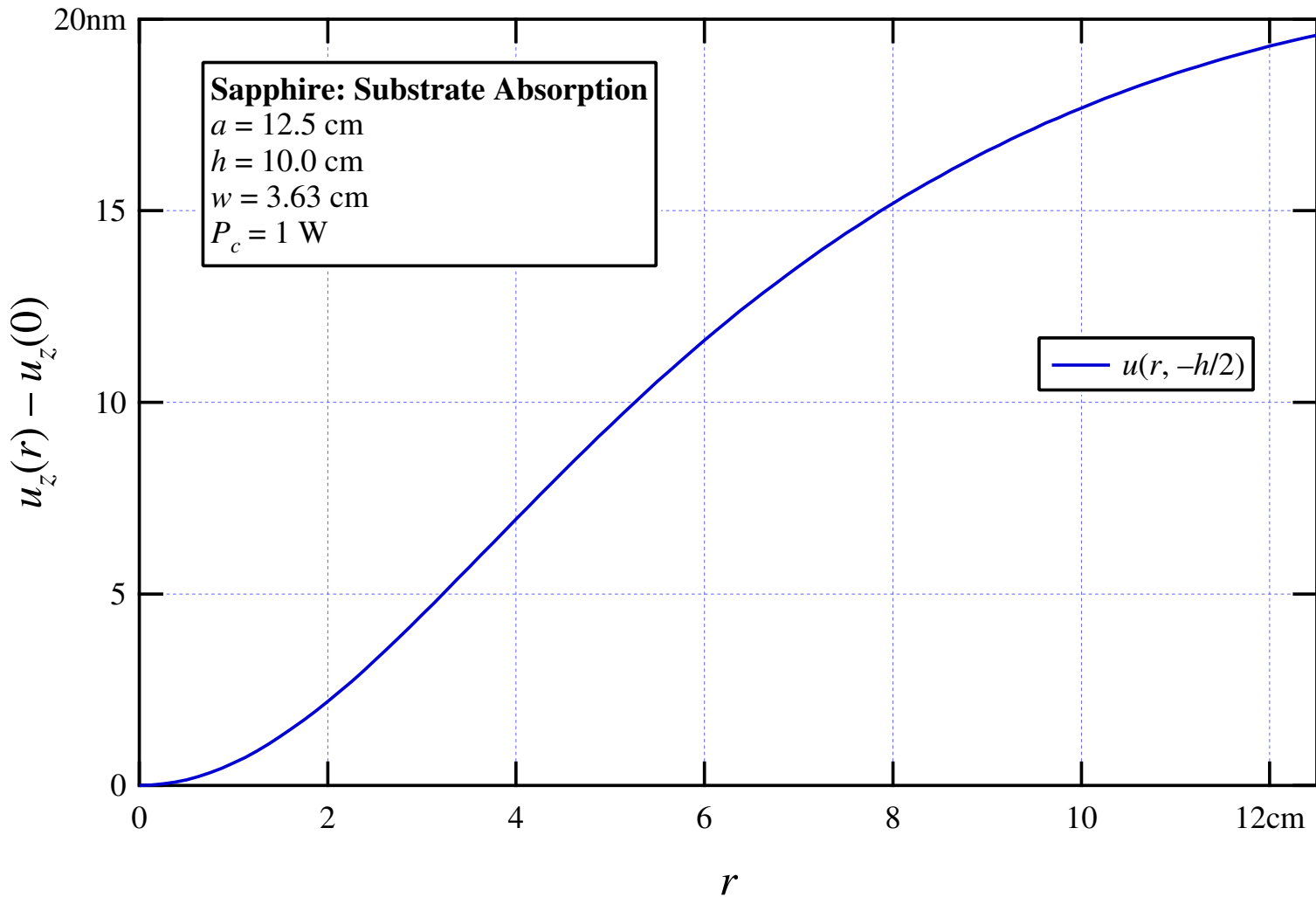
$$u_c \left( r, \pm \frac{h}{2} \right) = \mp P_c \frac{(1 + \nu) \alpha_T}{k_T} \sum_{k=0}^{\infty} \frac{a^2 p_k}{\zeta_k} [A_k \sinh(\gamma_k) \pm B_k \cosh(\gamma_k)] \\ \times J_0 \left( \zeta_k \frac{r}{a} \right) - \frac{1}{2} \frac{(1 - \nu) \alpha_T}{k_T} C_1 \left( \frac{r}{a} \right)^2$$

$$u_s \left( r, \pm \frac{h}{2} \right) = \mp P_s \frac{(1 + \nu) \alpha_T}{k_T} \sum_{k=0}^{\infty} \frac{a^2 p_k \sinh(\gamma_k)}{\zeta_k^2 \gamma_k} \\ \times \left[ \tau A_k - \frac{\sinh(\gamma_k)}{\gamma_k + \cosh(\gamma_k) \sinh(\gamma_k)} \right] J_0 \left( \zeta_k \frac{r}{a} \right)$$

# SAPPHIRE COATING ABSORPTION



# SAPPHIRE SUBSTRATE ABSORPTION



# THERMAL DEFORMATION OPERATOR

---

The propagation phase perturbation due to the surface OPD at the HR is

$$\phi(r, \pm h/2) = \frac{2\pi}{\lambda_0} \Delta n(z) u(r, \pm h/2)$$

where  $u(r, \pm h/2)$  is the *linear* sum of contributions from deformations due to absorption in both coatings (HR and AR) and the substrate.

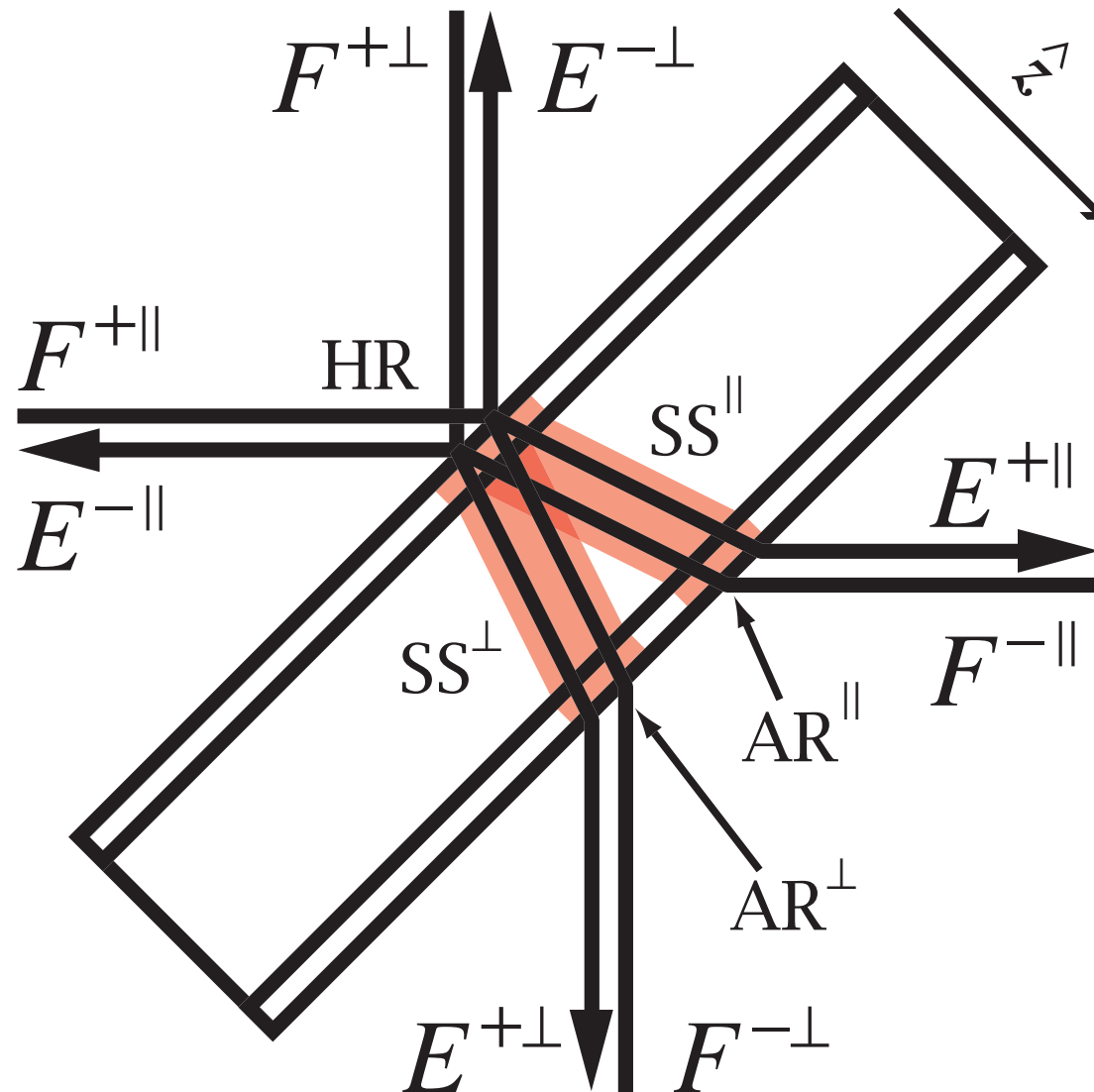
Matrix elements of the thermal deformation operator (unitary approx):

$$\begin{aligned} U_{m'n';mn} &= \iint_{-\infty}^{\infty} dx dy u_{m'n'}^\dagger(x, y) u_{mn}(x, y) e^{i\phi(r)} \\ &\equiv \exp \left[ i \iint_{-\infty}^{\infty} dx dy u_{m'n'}^\dagger(x, y) u_{mn}(x, y) \phi(r) \right] \end{aligned}$$

Since  $\phi(r) \propto r^2$ , TEM<sub>00</sub> is coupled to both TEM<sub>20</sub> and TEM<sub>02</sub>.

# BEAMSPLITTER THERMAL DISTORTIONS

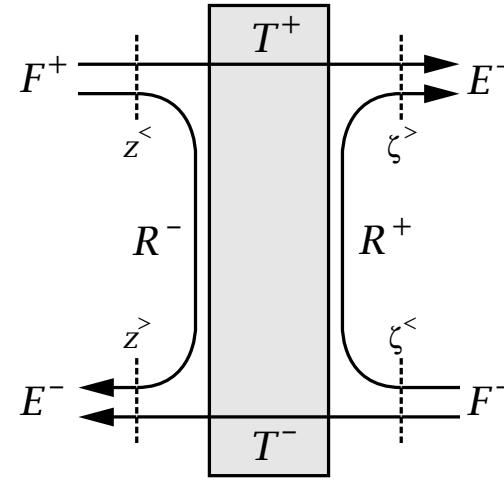
---



# MIRROR TRANSFER MATRIX

---

$$\begin{bmatrix} E^- \\ E^+ \end{bmatrix} = \begin{bmatrix} T^- & R^- \\ R^+ & T^+ \end{bmatrix} \begin{bmatrix} F^- \\ F^+ \end{bmatrix}$$



$$T^- = i t t_s (C^- C^+)^{1/2} S U A$$

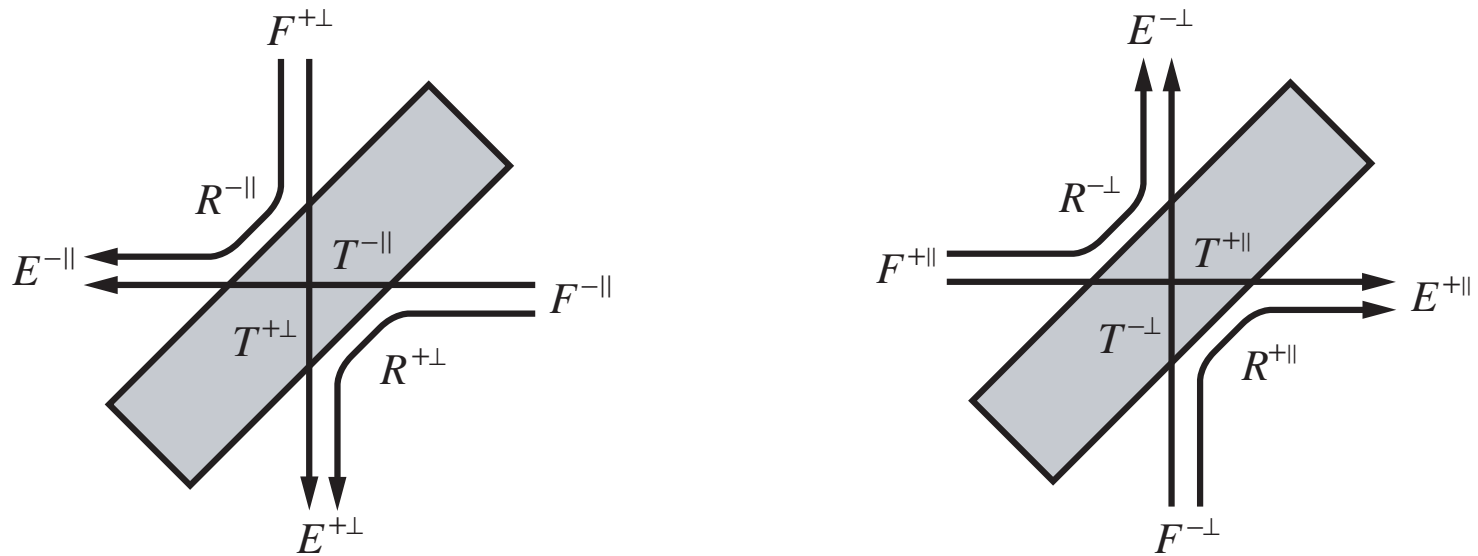
$$T^+ = i t t_s U S (C^+ C^-)^{1/2} A$$

$$R^- = -r e^{-i 2k \Delta z} C^- A, \text{ and}$$

$$R^+ = -r t_s^2 e^{+i 2k \Delta z} U S C^+ S U A$$

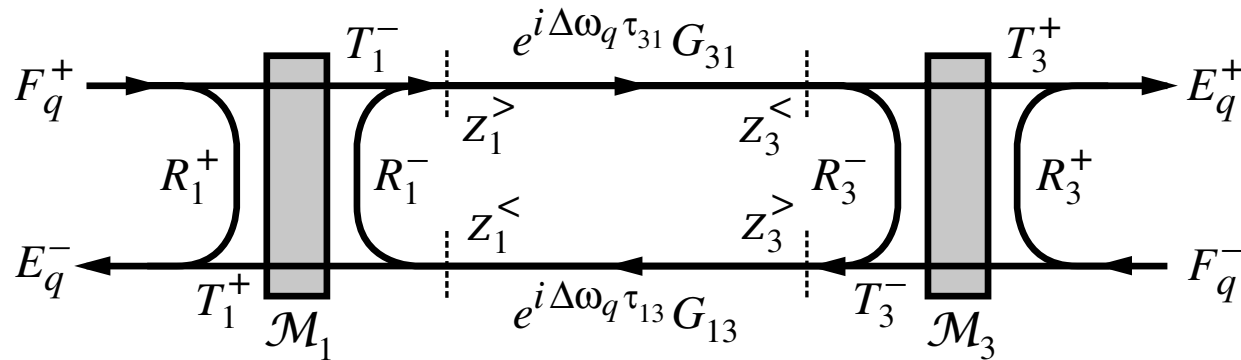


# BEAMSPLITTER TRANSFER MATRIX



$$\begin{bmatrix} E^{-\parallel} \\ E^{+\perp} \\ E^{+\parallel} \\ E^{-\perp} \end{bmatrix} = \begin{bmatrix} T^{-\parallel} & R^{-\parallel} & 0 & 0 \\ R^{+\perp} & T^{+\perp} & 0 & 0 \\ 0 & 0 & T^{+\parallel} & R^{+\parallel} \\ 0 & 0 & R^{-\perp} & T^{-\perp} \end{bmatrix} \begin{bmatrix} F^{-\parallel} \\ F^{+\perp} \\ F^{+\parallel} \\ F^{-\perp} \end{bmatrix}$$

# FABRY-PEROT INTERFEROMETER TRANSFER MATRIX



Transfer matrix:

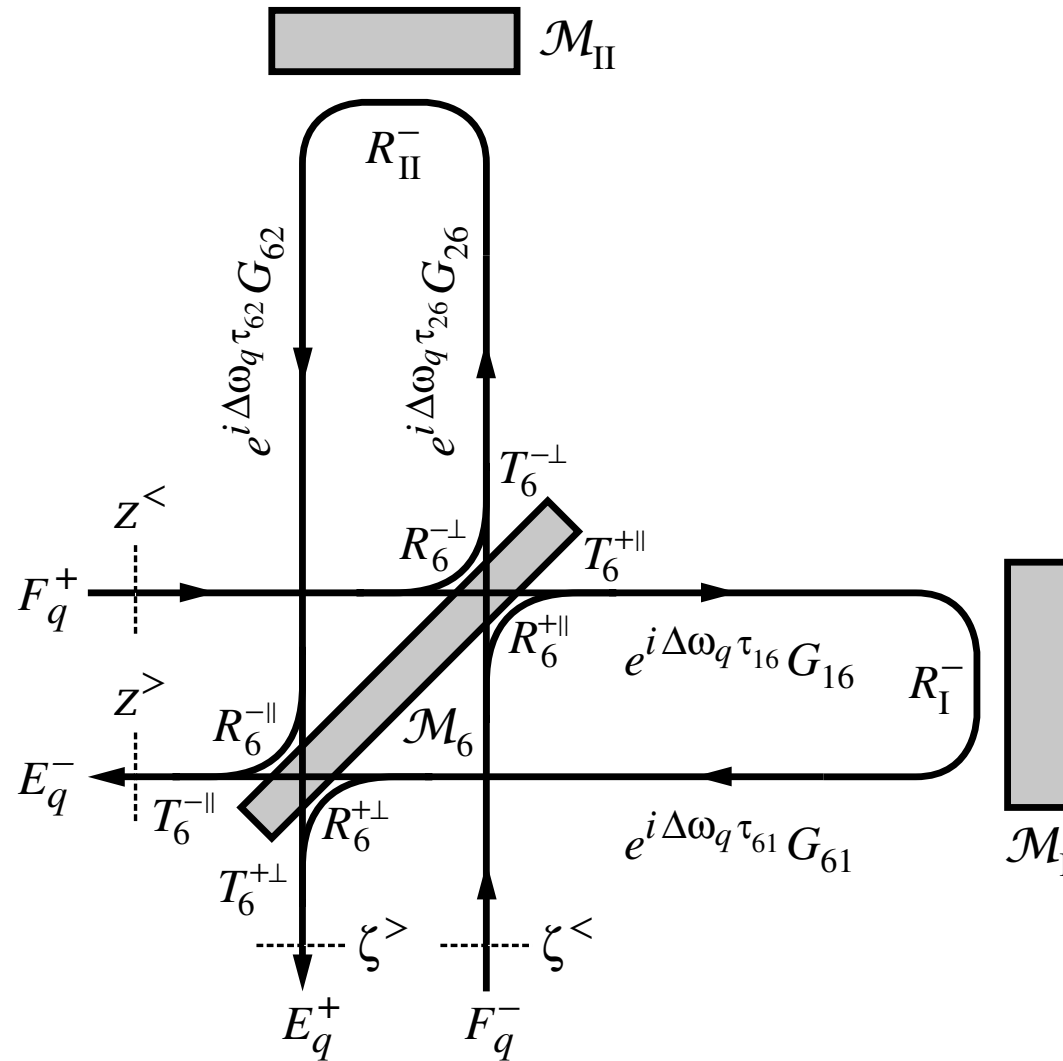
$$\begin{bmatrix} E_q^- \\ E_q^+ \end{bmatrix} = \begin{bmatrix} T_{\text{FPI}}^-(\Delta\omega_q) & R_{\text{FPI}}^-(\Delta\omega_q) \\ R_{\text{FPI}}^+(\Delta\omega_q) & T_{\text{FPI}}^+(\Delta\omega_q) \end{bmatrix} \begin{bmatrix} F_q^- \\ F_q^+ \end{bmatrix},$$

Example:

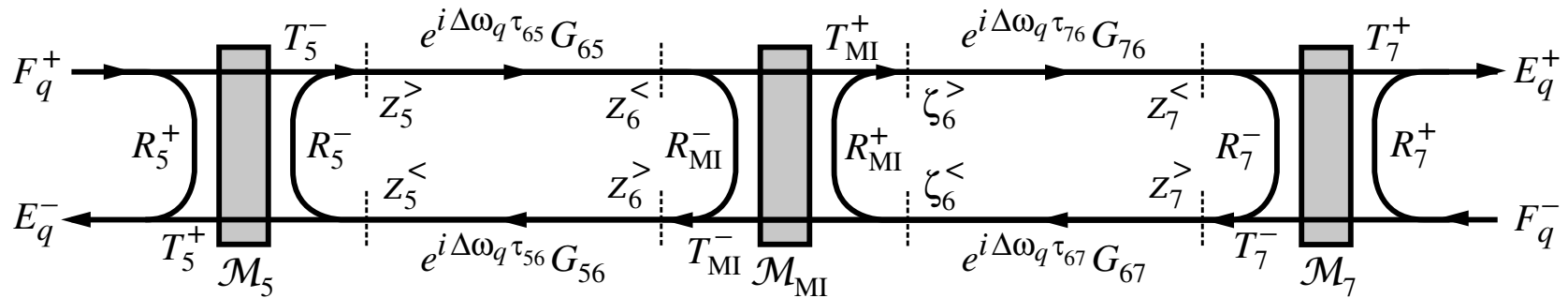
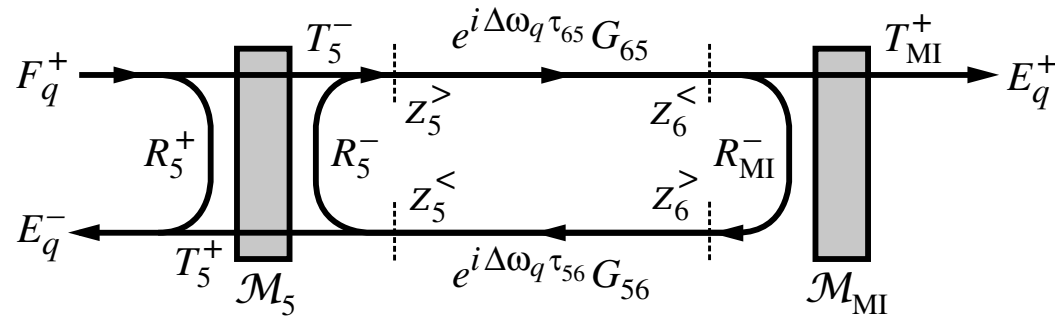
$$R_{\text{FPI}}^-(\Delta\omega) = R_1^+ + T_1^+ H_1^+(\Delta\omega)$$

$$H_1^+(\Delta\omega) = \left(1 - e^{i2\Delta\omega\tau_{13}} G_{13} R_3^- G_{31} R_1^-\right)^{-1} e^{i2\Delta\omega\tau_{13}} G_{13} R_3^- G_{31} T_1^-$$

# MICHELSON INTERFEROMETER TRANSFER MATRIX

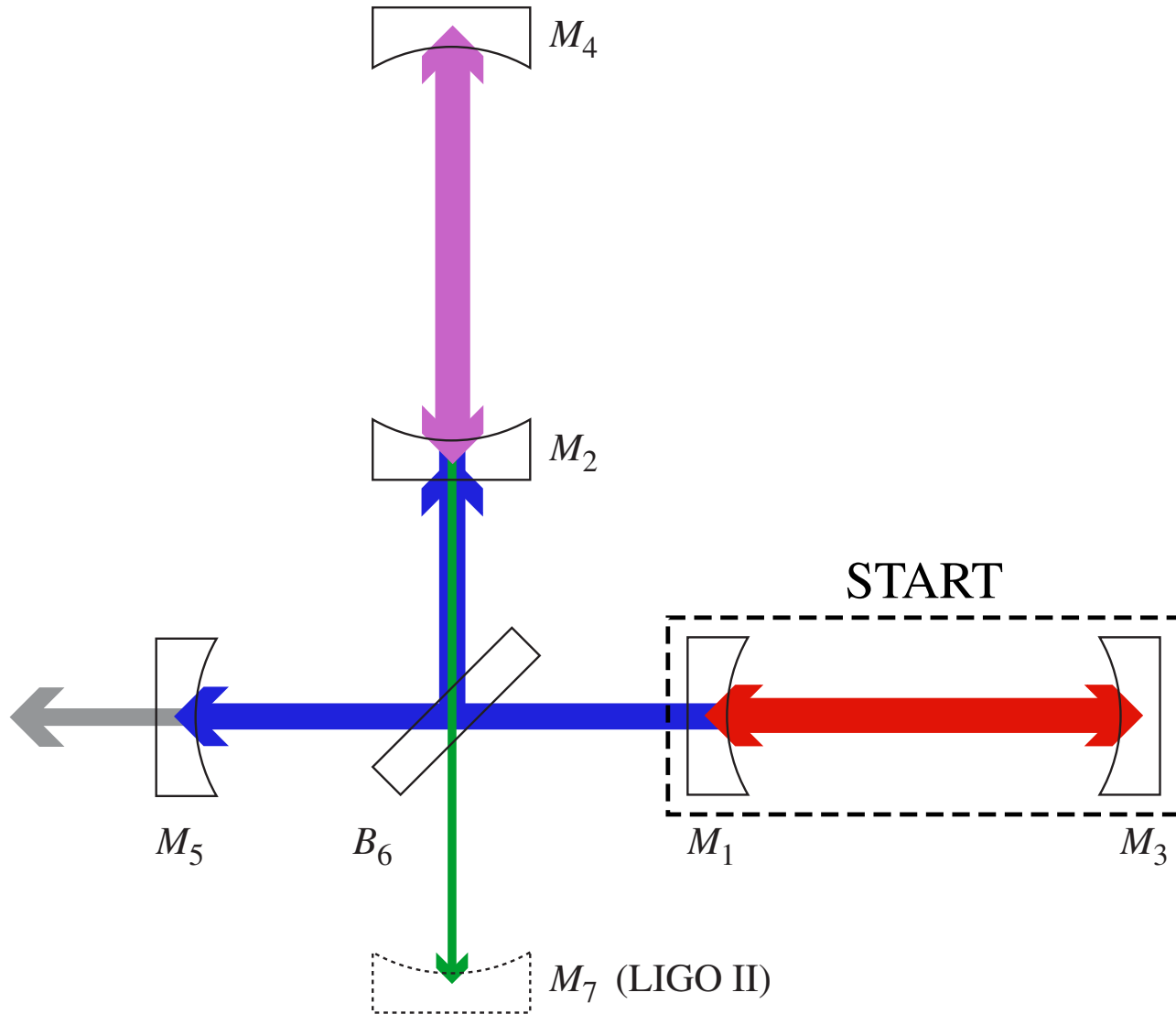


# LIGO INTERFEROMETER TRANSFER MATRIX



# IFO COUPLING SCHEMATIC

---



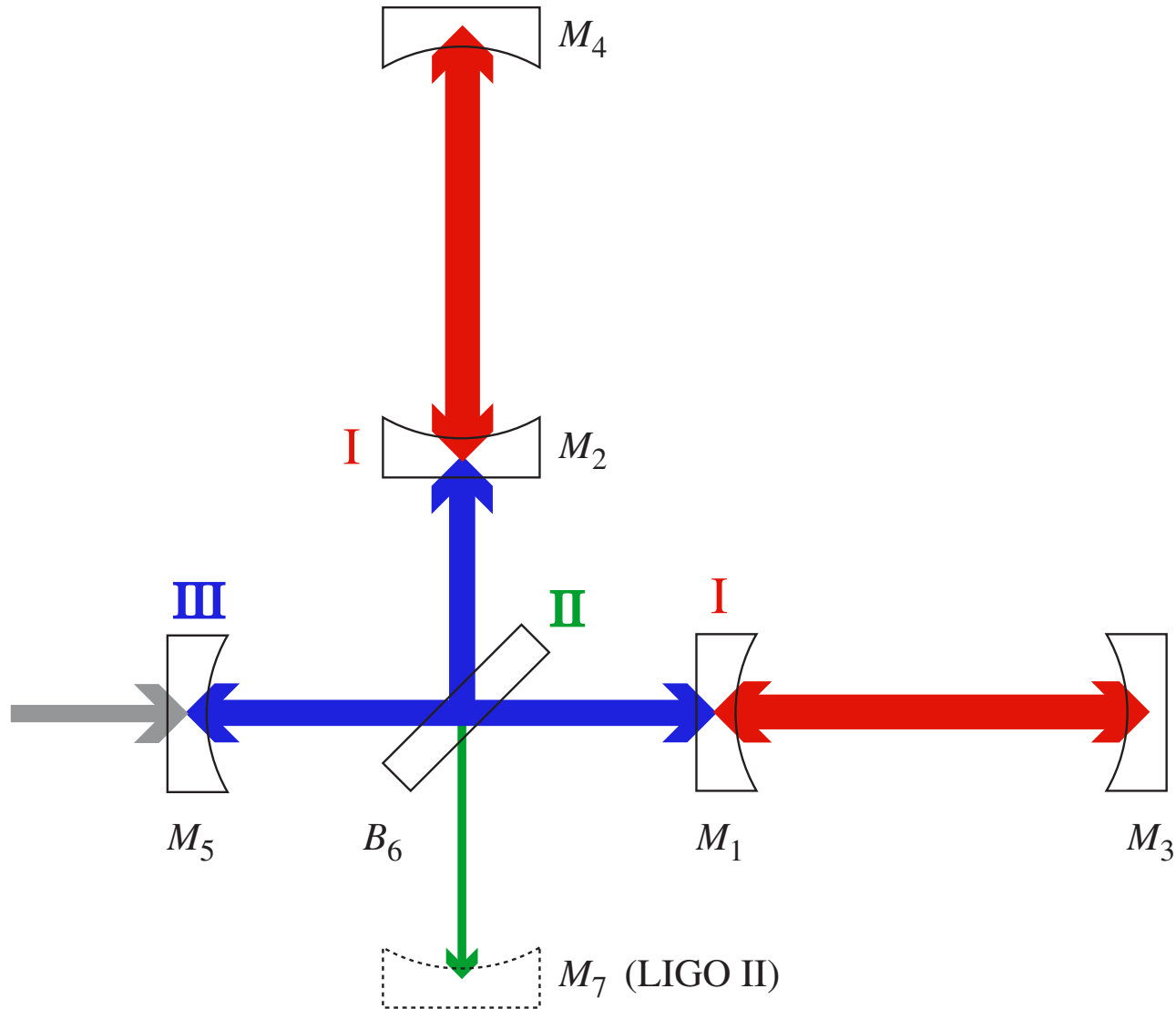
# MODEL OF IFO COUPLING

---

- Choose a *primary* FPI as a reference cavity; initial mirror properties define the “fundamental” unperturbed eigenmodes → basis functions
- Propagate the unperturbed basis functions from the FPI to the PRM and SRM; choose wavefront curvatures at mirror reference planes as unperturbed mirror curvatures
- Propagate basis to the secondary FPI; load FPI with fundamental basis
- Propagate out through the PRM to define a basis for the input field
- Provides a basis for the recycled fields even if recycling cavities are unstable

# RESONATOR LENGTH PSEUDOLOCKING

---



# RESONATOR LENGTH PSEUDOLOCKING

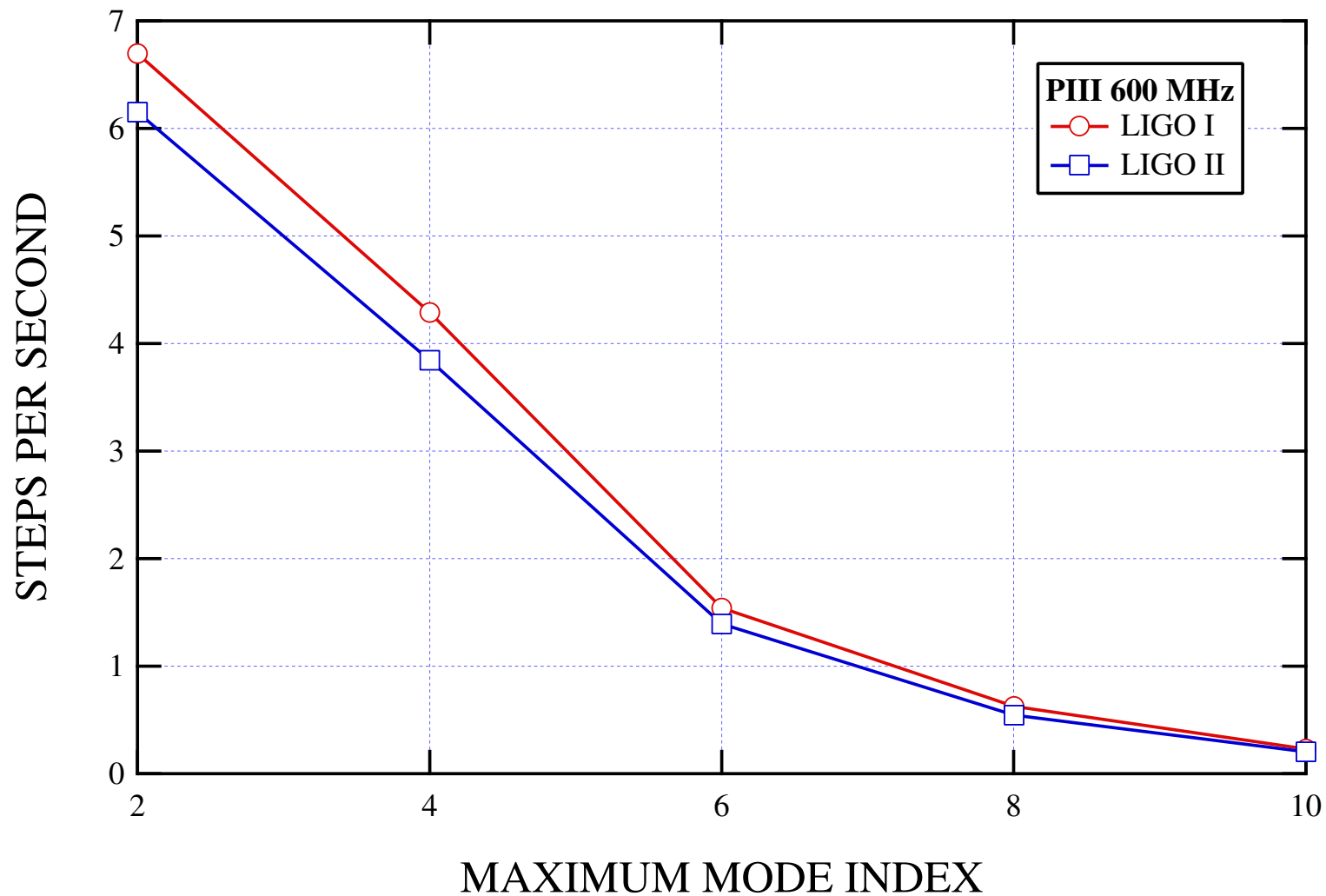
---

Self-contained simulations: implicit four-stage *pseudolocker*

1. *FPI* stage adjusts the positions of the FPI ITMs to maximize round-trip carrier TEM<sub>00</sub> enhancement.
2. *Dark Port* stage adjusts the beamsplitter position so that the amplitude of the carrier TEM<sub>00</sub> mode is minimized at the dark port.
3. *Power Recycling* stage adjusts the position of the PR mirror to maximize carrier TEM<sub>00</sub> enhancement.
4. *Signal Recycling* stage adjusts the position of the SR mirror to optimize carrier TEM<sub>00</sub> phase at the SR.

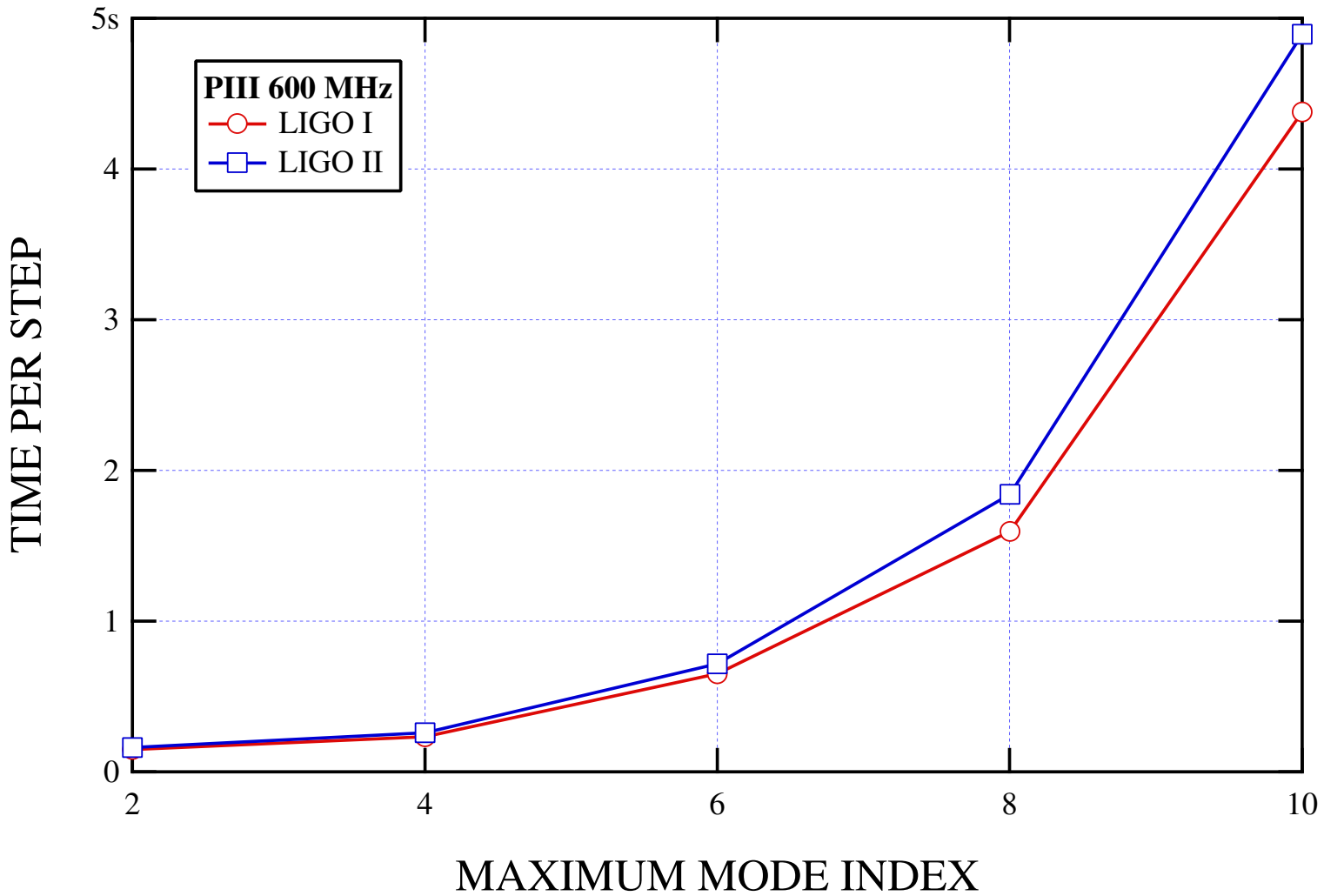


# MELODY STEP RATE

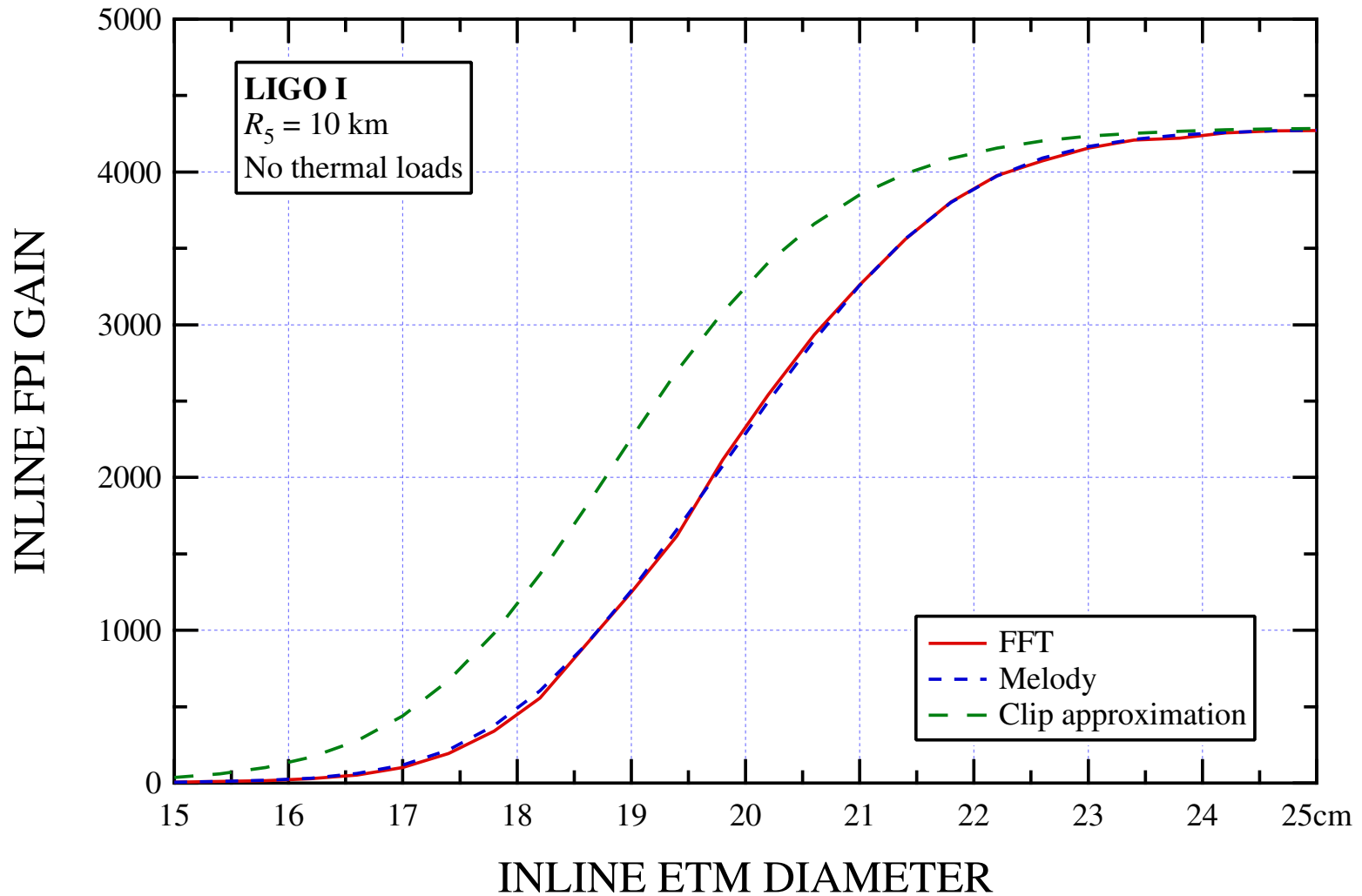


# MELODY STEP TIME

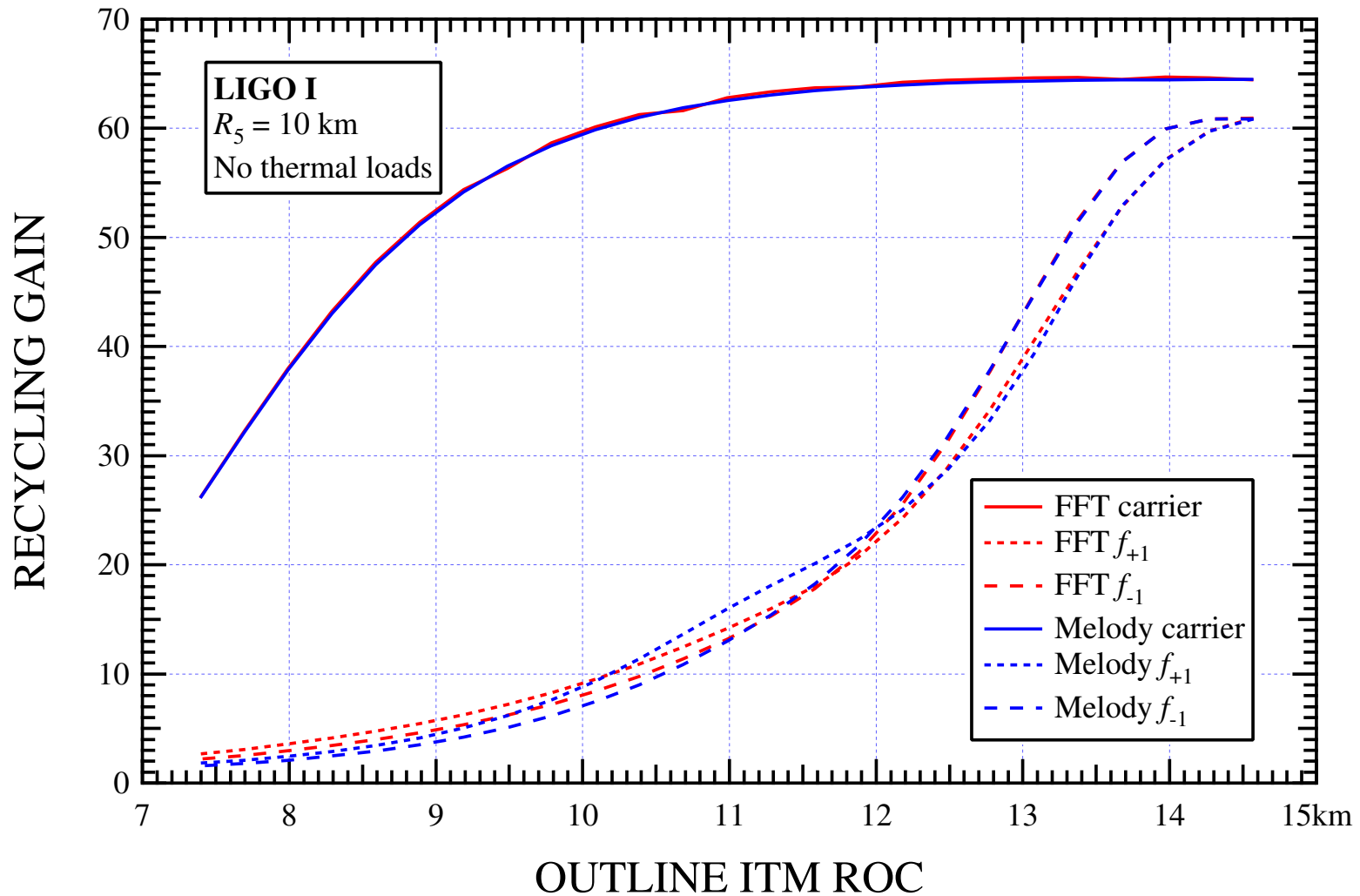
---



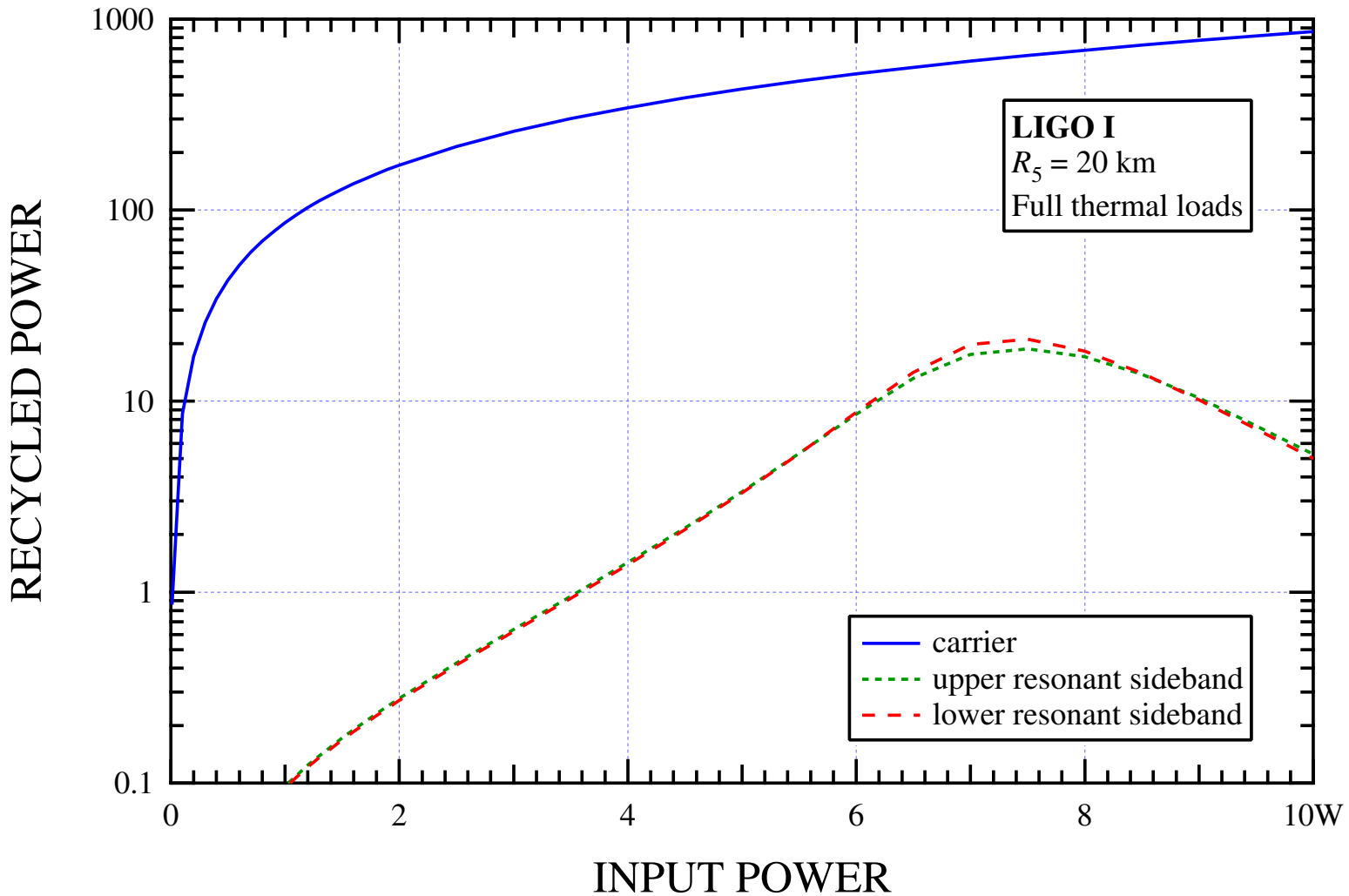
# MELODY/FFT APERTURE COMPARISON



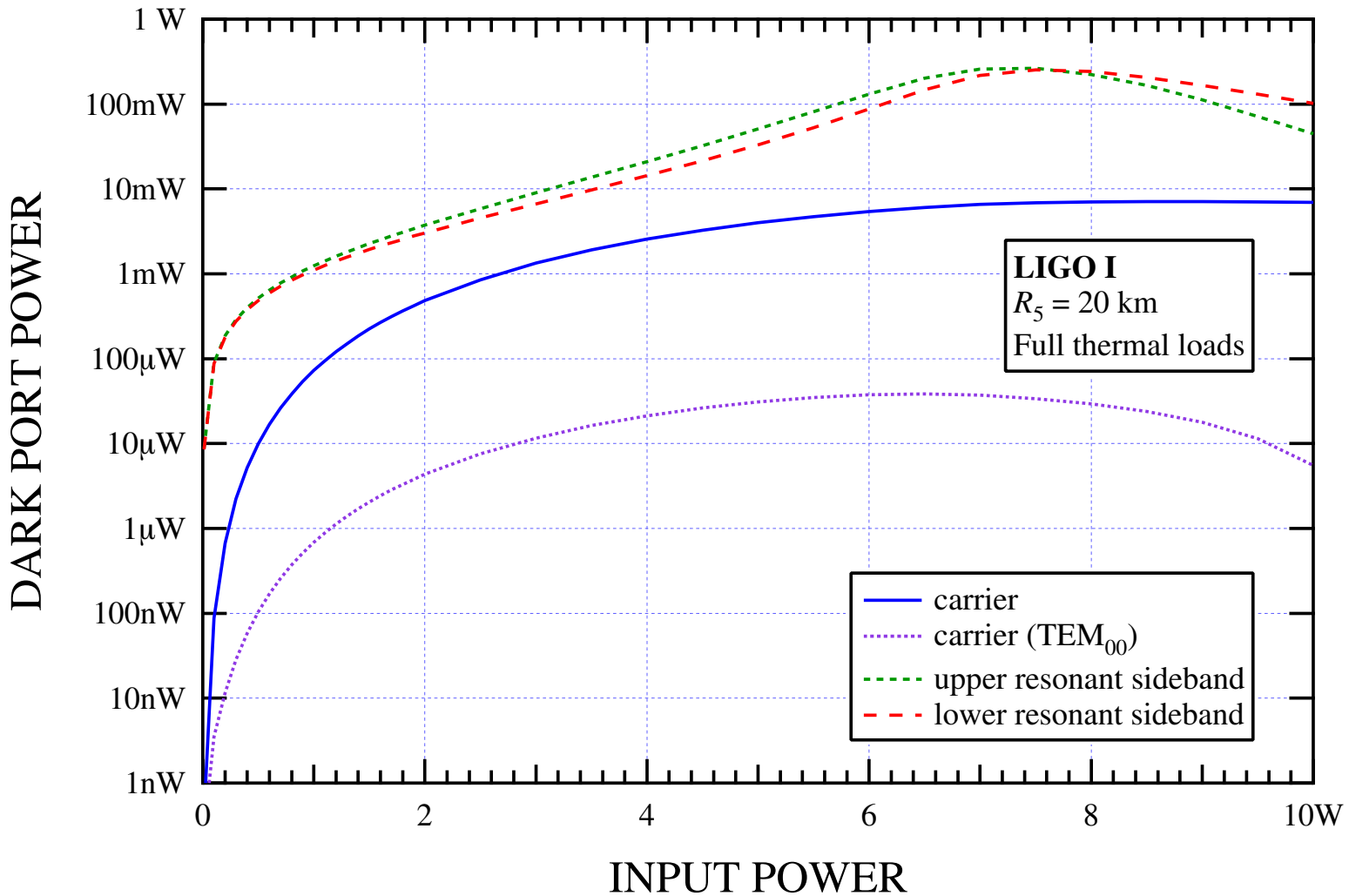
# MELODY/FFT ROC COMPARISON



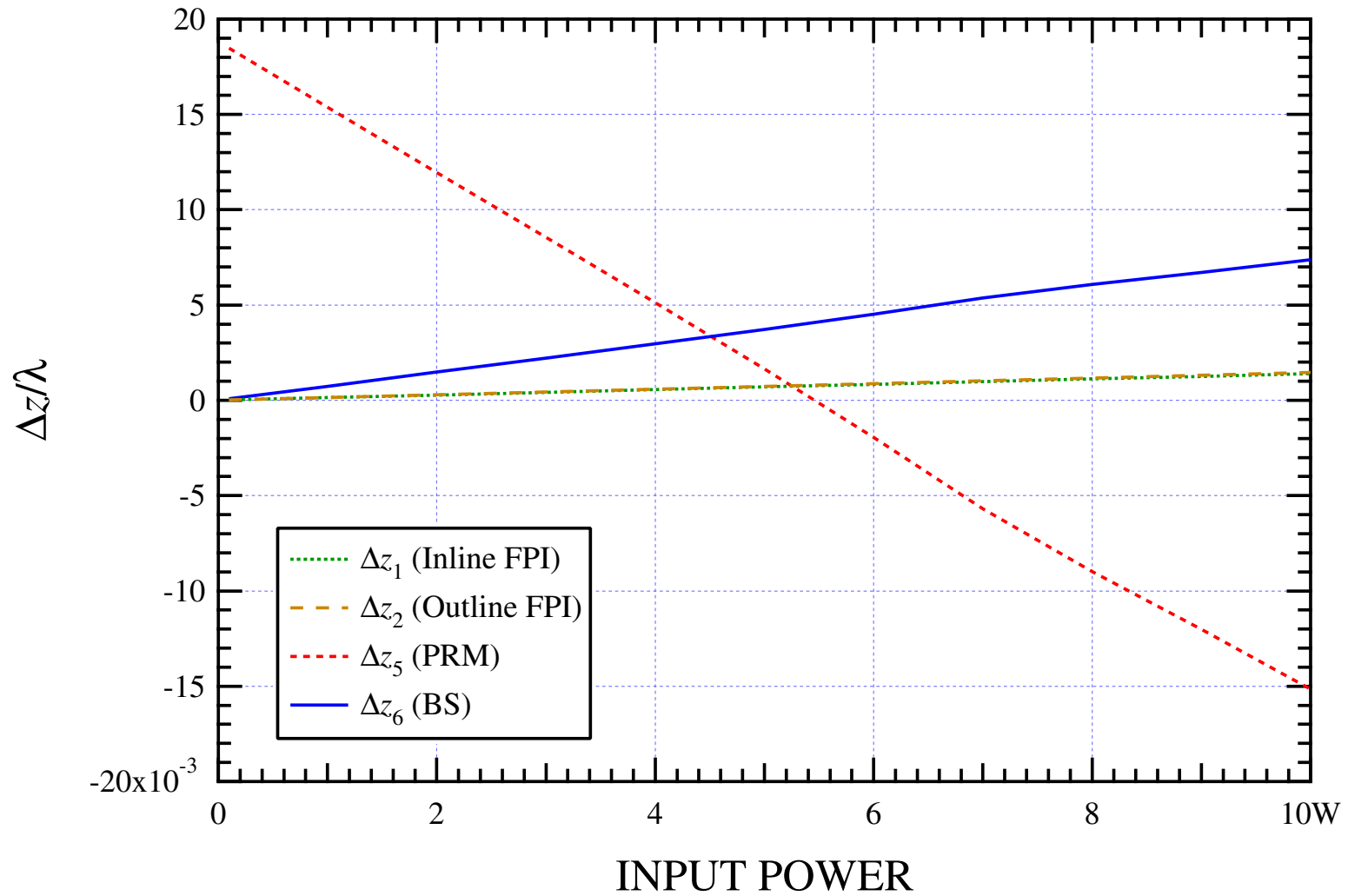
# LIGO I: HOT, $R_5 = 20$ km, $E_0 \equiv \text{TEM}_{00}$



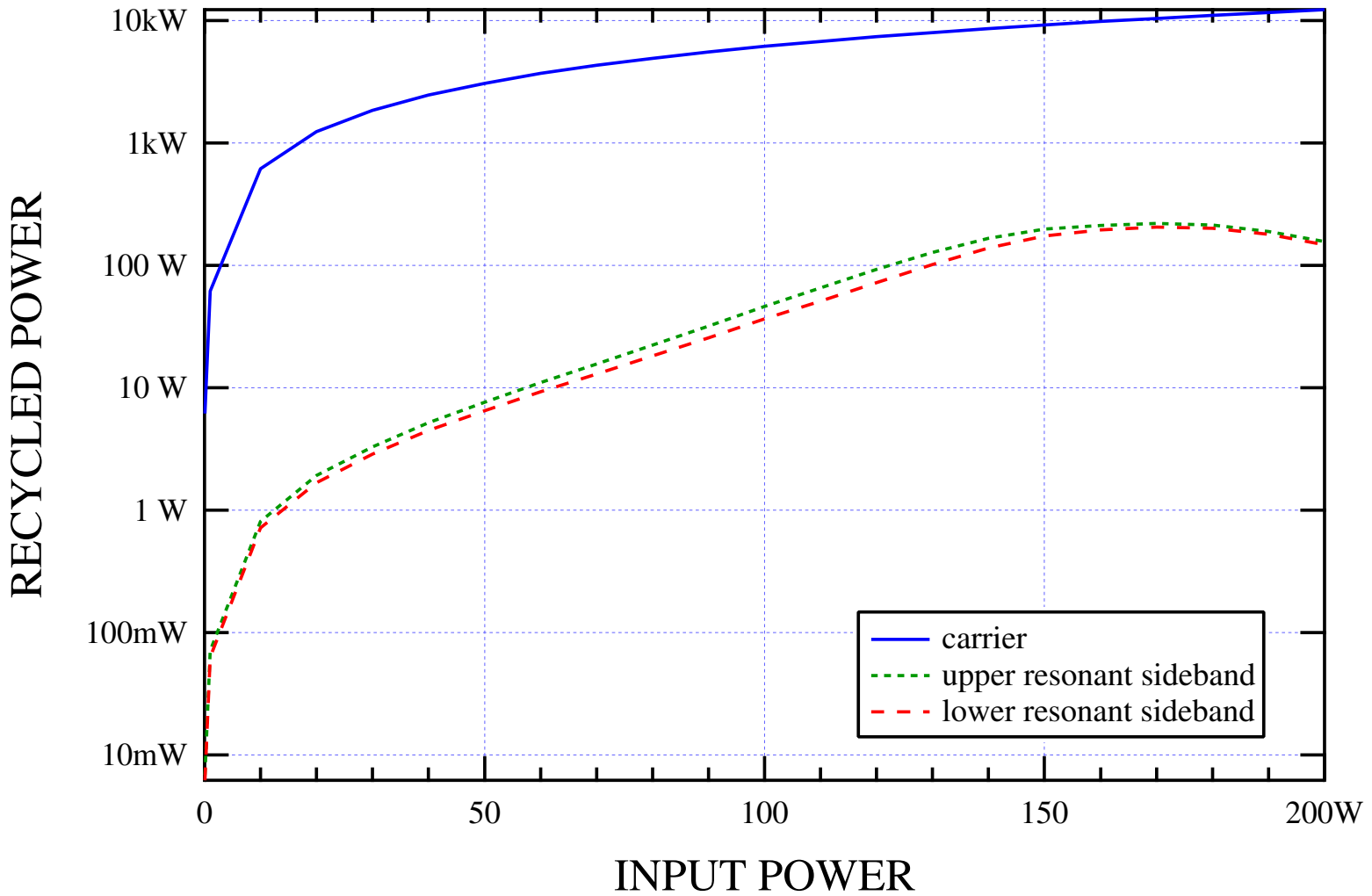
# LIGO I: HOT, $R_5 = 20$ km, $E_0 \equiv \text{TEM}_{00}$



# LIGO I PSEUDOLOCKER OUTPUT

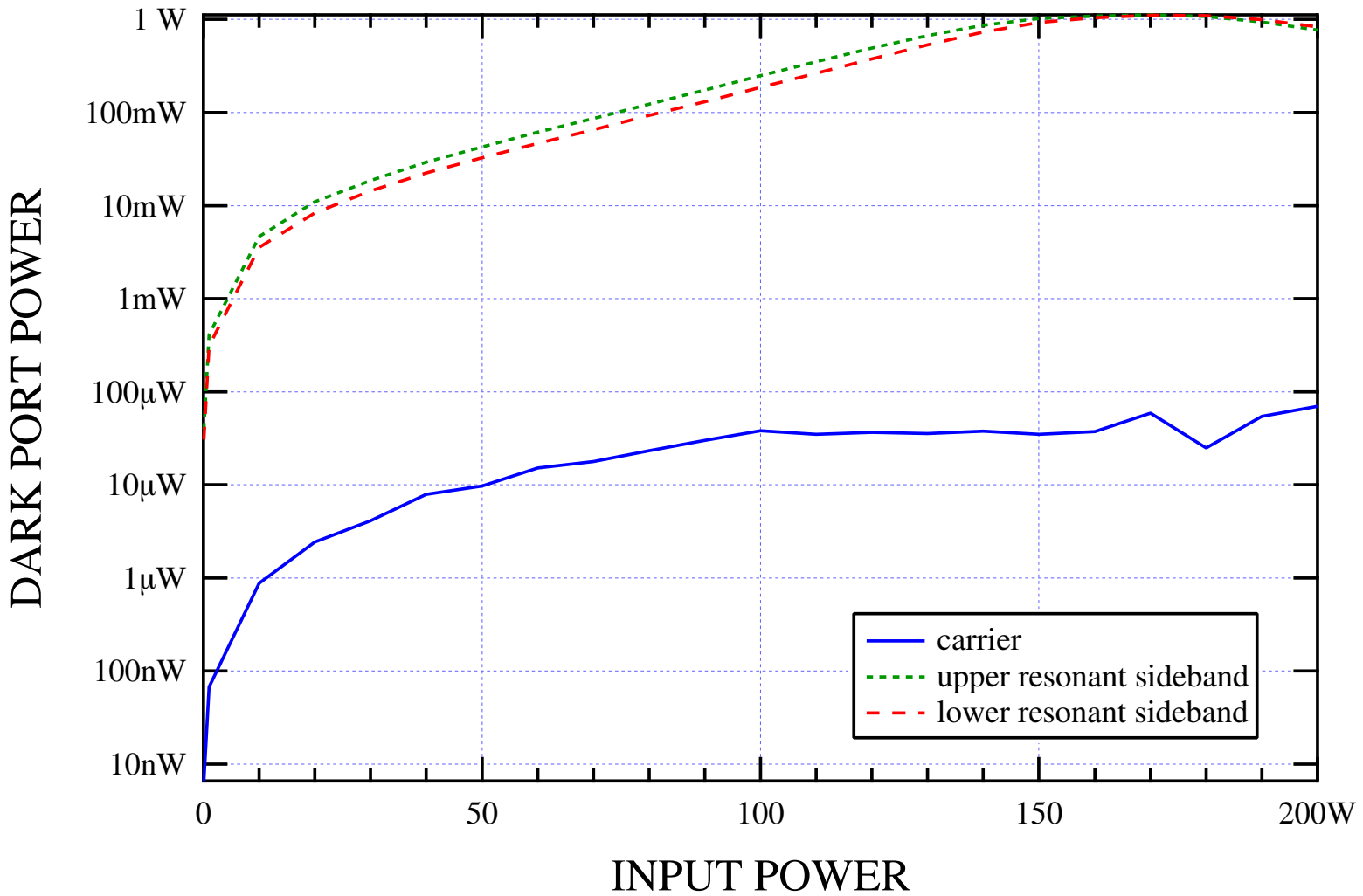


# LIGO II RECYCLED POWER

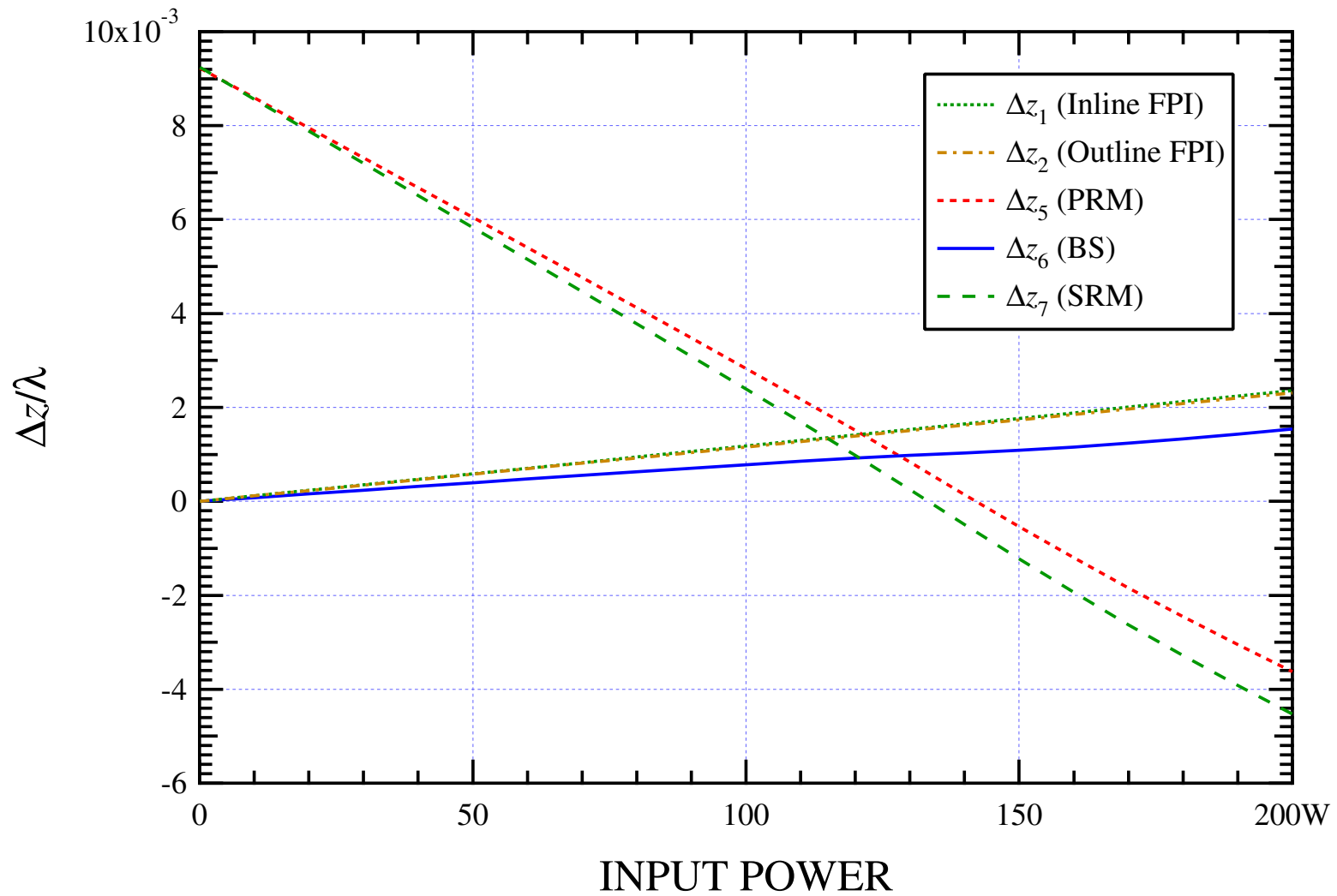




# LIGO II DARK PORT OUTPUT POWER



# LIGO II PSEUDOLOCKER OUTPUT



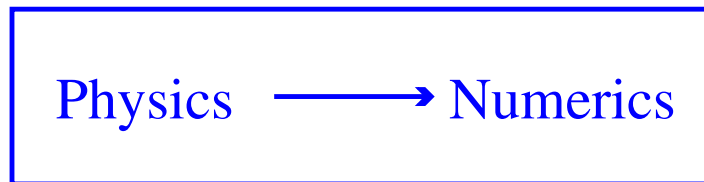
# MELODY/MATLAB FEATURES

---

- Simple object-oriented architecture in MATLAB
- Flexible modulation and resonance schemes
- Arbitrary number of spatial basis functions
- Aperture diffraction and mirror/field curvature mismatch
- Hello-Vinet mirror thermal lens and surface deformation
- Pseudolockers for LIGO I/II
- Astigmatic beamsplitter thermal lens
- Precomputation of all matrix operators available

# MELODY FEATURE LIFE-CYCLE

---



- Transient thermal loading
- Static thermal compensation



- Mirror misalignment



- Static thermal loading
- Static thermal deformation

# NEW FEATURE PRIORITIES

---

1. IFO eigenmode extraction for mode-matching
2. Add artificial differential gravitational-wave signal sidebands
3. Add compensation plates to basic configuration
4. Sideband optimization: common/differential lengths, Schnupp asymmetry, modulation frequency
5. Add curvature mismatch (with astigmatism) for tilted curved optics
6. Demodulation routine for detector class
7. Transient thermal loading
8. Mirror misalignment operators

# IMPLEMENTATION

---

- Use object-oriented programming in MATLAB: primitive classes, encapsulation, function/operator overloading, and inheritance
- Define classes for mirrors, Fabry-Perot and LIGO interferometers, electric fields (Hermite-Gauss, RF-modulated), and detectors
- Encapsulate classes representing simpler entities (mirrors, beamsplitters, laser fields) in classes representing interferometers
- Design simple class interfaces allowing calculations and simulations to be driven by MATLAB scripts

# IMPLEMENTATION SCHEMATIC

---

User-defined  
driver scripts

---

Melody architecture  
and classes (@mirror, ...)  
*“Qui ci sono dei mostri.”*

# SCRIPT-LEVEL FEATURES

---

- Input powers, modulation frequencies and depths
- All mirror parameters (e.g., thermal constants, orientation and micro-position)
- All interferometer cavity lengths
- Power/signal recycling
- Iteration and solution methods
- Graphics, object storage(!), post-processing
- Full interactive MATLAB functionality



# TWO-PHASE THERMAL/TEMPORAL SIMULATIONS

---

Characteristic time:  $t_c = \rho C a^2 / k_T \approx 5$  h for fused silica ( $a = 0.125$  m)

## THERMAL (Script-driven)

1. Run thermal relaxation code, including power-dependent optimizations (e.g., modulation depths, SRM reflectivity)
2. SAVE ligo object after stability is reached for each power level

## TEMPORAL (Script-driven, SIMULINK)

1. LOAD ligo object for a specified input power
2. Perturb mirrors and simulate temporal response

## SUBSET OF CLASSES

---

**laser\_field** Stores all spatial components for all operating sidebands, and the frequencies of those sidebands.

**mirror** Maintains all perturbation matrices (e.g., thermal and angular); encapsulates mirror parameters, two laser\_field objects, detectors.

**beamsplitter** Special case of mirror for 45° beamsplitter; uses numerical temperature distribution.

**detector** Demodulation detector array; almost complete.

**fpi** Fabry-Perot Interferometer

**ligo** LIGO I/II Interferometers

# LASER\_FIELD OBJECT DATA

---

basis: Hermite-Gauss

[f\0, f\1, -f\1, f\2, -f\2] = [0, 2.3971e+001, -2.3971e+001, 3.5956e+001, -3.5956e+001]

TEM_00	-7.36e-01	-2.45e-02i	-7.48e-02	-5.45e-04i	-7.48e-02	-4.79e-04i	-5.35e-02	-2.04e-03i	-5.35e-02	-7.46e-04i
TEM_10	0		0		0		0		0	
TEM_01	0		0		0		0		0	
TEM_20	-1.64e-04	+1.86e-04i	8.15e-03	-9.54e-03i	7.54e-03	-9.98e-03i	-6.69e-07	+6.32e-06i	-9.64e-08	+6.47e-06i
TEM_11	0		0		0		0		0	
TEM_02	-1.64e-04	+1.86e-04i	8.15e-03	-9.54e-03i	7.54e-03	-9.98e-03i	-6.69e-07	+6.32e-06i	-9.64e-08	+6.47e-06i

- laser\_field class consists of data fields (*members*) and routines which operate on those fields
- Routines fall into two broad categories:
  - procedures** which alter the internal state of the object but do not return results (e.g., object update procedures)
  - functions** which return results but do not alter the internal state of the object (e.g., overloaded arithmetic operators)

# SIDEBAND REPRESENTATION

---

Define the propagation vector

$$k = k_0 + \Delta k_q,$$

where  $\Delta k_q/k_0 \ll 1$ ,  $\omega_0 \equiv k_0 c$ , and  $\Delta\omega_q \equiv \Delta k_q c$ . Write the time-dependent length as

$$L(t) \equiv L_0 + \Delta L(t),$$

where  $2k_0L_0 - \varphi_{00} = 2N\pi$  and  $\Delta L(t) \approx \lambda = 2\pi/k_0$ . Then

$$\begin{aligned} e^{i[2kL(t) - \varphi_{00}]} &= e^{i(2k_0L_0 - \varphi_{00})} e^{i[2k_0\Delta L(t)]} e^{i(2\Delta\omega_q L_0/c)} e^{i[2\Delta k_q \Delta L(t)]} \\ &= e^{i[2k_0\Delta L(t) + \Delta\omega_q \tau_0]} \end{aligned}$$

Include  $\Delta L(t)$  in mirror class; implement  $\Delta\omega_q \tau_0$  as a diagonal propagation matrix.

# FPI OBJECT UPDATE PROCEDURE

---

```
% Get the total field propagating away from the
% vacuum-coating interface of m_1, and then
% propagate that field to the vacuum-coating
% interface of m_2. This is the new 'front
% field' of m_2.
e_1_r = get_field(m_1, 'front');
e_2 = fp.gouy_prop * e_1_r * fp.kz_prop;
set_field(m_2, e_2, 'front');

% Get the total field propagating away from the
% vacuum-coating interface of m_2, and then
% propagate that field to the vacuum-coating
% interface of m_1. This is the new 'front
% field' of m_1.
e_2_r = get_field(m_2, 'front');
e_1 = fp.gouy_prop * e_2_r * fp.kz_prop;
set_field(m_1, e_1, 'front');
```

# LASER\_FIELD MTIMES FUNCTION

---

```
function e_3 = mtimes(e_1, e_2)
%
...
%
if isa(e_1, 'laser_field') & ~isa(e_2, 'laser_field')
% Initialize the structure e_3 with the same basis and sidebands
% as e_1, and multiply (matrix, using *) the elements of the
% matrix e_2 by the components of e_1.
    e_3.basis = e_1.basis;
    e_3.sideband = e_1.sideband;
    e_3.component = e_1.component*e_2;
elseif ~isa(e_1, 'laser_field') & isa(e_2, 'laser_field')
% Initialize the structure e_3 with the same basis and sidebands
% as e_2, and multiply (matrix, using *) the components of
% e_2 by the elements of the matrix e_1.
    e_3.basis = e_2.basis;
    e_3.sideband = e_2.sideband;
    e_3.component = e_1*e_2.component;
else
    error('Matrix multiplication of two laser_field objects is not allowed.');
```

```
end

% Create a new laser_field object from the struct e_3.
e_3 = class(e_3, 'laser_field');
```

## MATLAB/OOP REFERENCES

---

- Duane Hanselman and Bruce Littlefield, **Mastering MATLAB 6: A Comprehensive Tutorial and Reference** (Prentice-Hall, 2001); ISBN 0-13-019468-9
- Bertrand Meyer, **Object-Oriented Software Construction**, Second Edition (Prentice-Hall, 1997); ISBN 0-13-629155-4
- Paul F. Dubois, **Object Technology for Scientific Computing: Object-Oriented Numerical Software in Eiffel and C** (Prentice-Hall, 1997); ISBN 0-13-257808-X
- John J. Barton and Lee R. Nackman, **Scientific and Engineering C++** (Addison-Wesley, 1995); ISBN 0-201-53393-6