

## Chapter 8

### Summary

We have demonstrated that diffusive light can be used to image absorption, scattering and fluorescence lifetime and concentration in thick, turbid media. We hope we have convinced the reader that this work has important medical applications.

There is a great deal of work to be done if DPDW imaging is to become an accepted medical tool. First and foremost, we need many more studies which accurately measure the optical changes which accompany changes in the physiological state of tissues. The intrinsic heterogeneity of the body makes it difficult to measure *in vivo* the absorption and scattering coefficients. When tissue samples are studied *in vitro*, the blood and oxygen supply is cut off, thus changing the tissue environment. It is not clear whether or not *in vitro* measurements accurately reflect the optical properties of the sample *in vivo*.

We believe that DPDW imaging should be used to complement other imaging modalities. For example, a DPDW probe could be easily attached to an ultrasound, x-ray, or MRI imaging device as we saw in section 4.12. When the data from these two probes are used simultaneously, we can derive accurate structural information as well as quantitative measurements of the optical properties.

DPDW imaging can also be used for low resolution breast tumor screening. There are several academic and industrial research groups performing clinical evaluations of optical mammography systems [13, 81, 82]. These studies are particularly exciting because optical mammography is inexpensive, and has no known adverse side effects. The preliminary results of these studies are encouraging, but a great deal of work still

needs to be done.

## Appendix A

### Singular Matrices

A singular matrix is one which maps a vector, (other than the zero vector) to the zero vector. For example, the matrix

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad (\text{A.1})$$

maps any vector of the form  $\vec{\alpha} = (\alpha, \alpha)$  to the zero vector. This has very important implications for the inverse solution. Suppose we are trying to invert the following equation,

$$A\vec{x} = \vec{b}. \quad (\text{A.2})$$

It is clear that for any solution  $\vec{x}$ ,  $\vec{x} + \vec{\alpha}$  is also a solution, i.e. if

$$A\vec{x} = \vec{b} \quad (\text{A.3})$$

and

$$A\vec{\alpha} = \vec{0} \quad (\text{A.4})$$

then

$$A(\vec{b} + \vec{\alpha}) = A\vec{b} + A\vec{\alpha} \quad (\text{A.5})$$

$$= A\vec{b} + \vec{0} = A\vec{b} \quad (\text{A.6})$$

Thus for a singular matrix, there is not a unique solution. Mathematically, a singular matrix expresses itself as having a zero eigenvalue. In numerical calculations, an

eigenvalue is rarely exactly equal to zero, but is *numerically zero*. Numerically zero means that the value of the number is below the numerical precision of the computer.

## Appendix B

### Time Resolved Spectroscopy

A schematic of the time resolved spectroscopy system (TRS) is shown in figure B.1. The TRS is a single photon counting device built by Hamamatsu Corporation, Japan. A pulsed laser diode with a full-width half-max of about 50 ps and 5 MHz repetition rate is coupled to a 200  $\mu\text{m}$  multi-mode optical fiber. The fiber delivers the light to the sample, and a 3 mm optical fiber bundle is used to collect the light from the sample. This second fiber delivers the collected light to a microchannel plate photomultiplier tube (MCP-PMT)

When a photon strikes the photomultiplier tube, a current is generated, and passed into a constant fraction discriminator (CFD). The CFD converts a irregularly shaped pulse into a clean (TTL) step voltage (figure B.2). This step voltage is passed to the time amplitude converter (TAC). The TAC is a simple timing device which is triggered by the same trigger which pulses the laser source. The timing is stopped when the TAC receives a signal from the CFD. An equivalent circuit is shown in figure B.3. A voltage which is proportional to the amount of time between the trigger and the pulse from the CFD is output from the TAC, to an analog to digital (A/D) converter. Note that only the first photon per pulse will be counted in this system. For this reason, the source light is attenuated so that no more than one photon per pulse is detected.

The A/D converter passes a digital number corresponding to the voltage from the TAC to a multichannel analyzer (MCA). The MCA collects the digital voltage signals and collects them into corresponding time bins which are stored in the computer. The

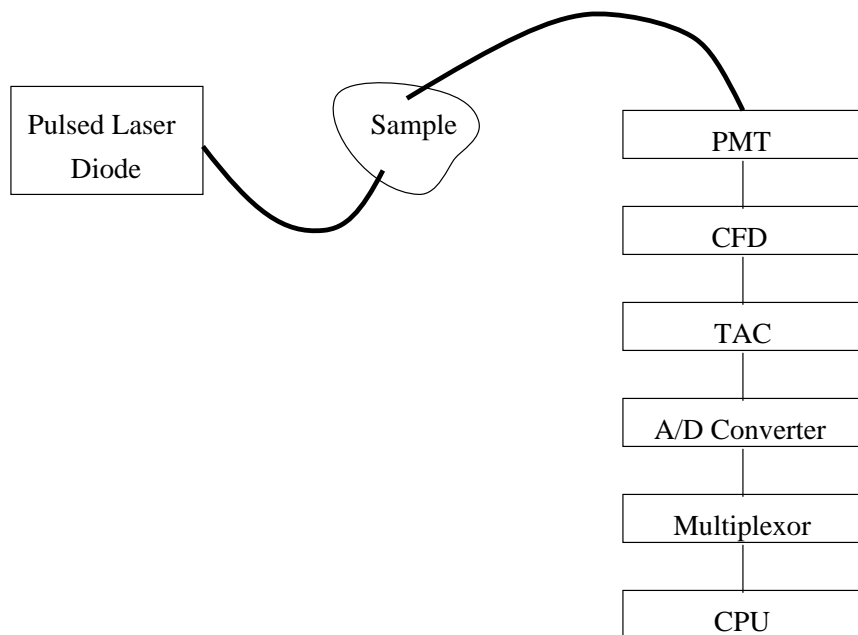


Figure B.1: A schematic of the TRS system

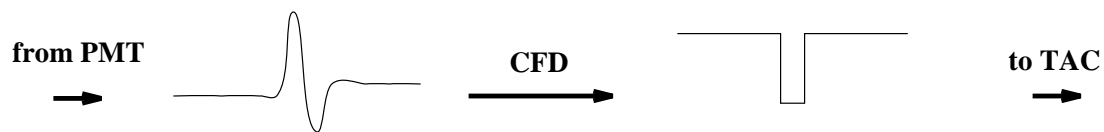


Figure B.2: Signal processing by the constant fraction discriminator (CFD)

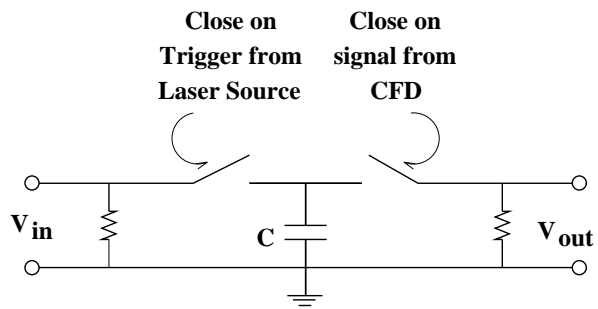


Figure B.3: Equivalent circuit for the time amplitude converter (TAC).

time per bin is calibrated by measuring two signals displaced by a known time delay.

For each pulse, we collect a single photon, and add a count to one bin. This process is repeated for millions of pulses (typically we use a 30 second integration time and a 5 MHz pulse rate) and a histogram of number of photons per time bin is generated. For examples of TRS data and algorithms for fitting the optical properties of unknown media, see Patterson *et al.* [3].





## Appendix C

### Time Domain Fluorescent DPDW Derivation

We will follow a time domain discussion similar to that of Sevick *et al.* [21, 67] and take the Fourier transform of the results to obtain the frequency domain results.

If we inject a pulse of photons at a time, 0 at a position  $\mathbf{r}_s$ , into a highly scattering medium, then according to the diffusion equation the number of photons per unit volume,  $u$ , reaching a position  $\mathbf{r}$  at a time  $t'$  is [3]

$$u(\mathbf{r}_s, \mathbf{r}, t') = \frac{\exp(-\frac{|\mathbf{r}-\mathbf{r}_s|^2}{4Dvt'} - \mu_a vt')}{(4\pi Dvt')^{3/2}}. \quad (\text{C.1})$$

A fluorophore at  $\mathbf{r}$  will absorb and re-radiate the light at a time  $t^*$  with some efficiency  $\eta$ , and a the rate described by the characteristic lifetime  $\tau$ ,

$$\frac{\eta}{\tau} \int_0^{t^*} dt' u(\mathbf{r}_s, \mathbf{r}, t') e^{(t'-t^*)/\tau}. \quad (\text{C.2})$$

This is now our fluorescent source term. Photons from this source will propagate from the fluorophore, to the detector and arrive at some time  $t$ . So the detected signal is

$$u_{fl}(\mathbf{r}_s, \mathbf{r}, \mathbf{r}_d, \tau, t) = \frac{\eta}{\tau} \int_0^t dt^* \int_0^{t^*} dt' u(\mathbf{r}_s, \mathbf{r}, t') \exp(\frac{t'-t^*}{\tau}) u'(\mathbf{r}_d, \mathbf{r}, t-t^*). \quad (\text{C.3})$$

We have used the symbol  $u'$  instead of  $u$  as a reminder that the propagation from the fluorophore to the detector is a function of the optical properties of the medium at the *fluorescent* wavelength.

To convert from the time domain equation to the frequency domain equation, we first change the limits of the integrals

$$u_{fl}(\mathbf{r}_s, \mathbf{r}, \mathbf{r}_d, \tau, t) = \frac{\eta}{\tau} \int_0^\infty dt^* \int_{-\infty}^{t^*} dt' u(\mathbf{r}_s, \mathbf{r}, t') \exp(\frac{t'-t^*}{\tau}) u'(\mathbf{r}_d, \mathbf{r}, t-t^*) \quad (\text{C.4})$$

since  $u(\mathbf{r}_s, \mathbf{r}, t') = 0$  when  $t' < 0$  and  $u'(\mathbf{r}_d, \mathbf{r}, t - t^*) = 0$  when  $t - t^* < 0$ . Next, we replace  $u$  and  $u'$  with their Fourier transforms,

$$u(\mathbf{r}_s, \mathbf{r}, t') = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega U(\mathbf{r}_s, \mathbf{r}, \omega) e^{-i\omega t'} = \frac{e^{ikr}}{4\pi Dr}, \quad (\text{C.5})$$

$$u_{fl} = \frac{\eta}{4\pi^2\tau} \int_0^{\infty} dt^* \int_{-\infty}^{t^*} dt' \int_{-\infty}^{\infty} d\omega \int_{-\infty}^{\infty} d\omega' \left\{ U(\mathbf{r}_s, \mathbf{r}, \omega) e^{-i\omega t'} \right. \\ \left. \times e^{\frac{t'-t^*}{\tau}} U'(\mathbf{r}_d, \mathbf{r}, \omega') e^{-i\omega'(t-t^*)} \right\}. \quad (\text{C.6})$$

Now we do the integral over  $dt'$

$$u_{fl} = \frac{\eta}{4\pi^2} \int_0^{\infty} dt^* \int_{-\infty}^{\infty} d\omega \int_{-\infty}^{\infty} d\omega' U(\mathbf{r}_s, \mathbf{r}, \omega) \frac{e^{-i\omega t^*}}{1 - i\omega\tau} U'(\mathbf{r}_d, \mathbf{r}, \omega') e^{-i\omega'(t-t^*)}, \quad (\text{C.7})$$

and then over  $dt^*$ ,

$$u_{fl}(\mathbf{r}_s, \mathbf{r}, \mathbf{r}_d, \tau, t) = \frac{\eta}{2\pi} \int_{-\infty}^{\infty} d\omega \int_{-\infty}^{\infty} d\omega' U(\mathbf{r}_s, \mathbf{r}, \omega) \frac{\delta(\omega - \omega')}{1 - i\omega\tau} U'(\mathbf{r}_d, \mathbf{r}, \omega') e^{-i\omega t}. \quad (\text{C.8})$$

Finally, integrating over  $\omega'$  we arrive at

$$u_{fl}(\mathbf{r}_s, \mathbf{r}, \mathbf{r}_d, \tau, t) = \frac{\eta}{2\pi} \int_{-\infty}^{\infty} d\omega U(\mathbf{r}_s, \mathbf{r}, \omega) \frac{1}{1 - i\omega\tau} U'(\mathbf{r}_d, \mathbf{r}, \omega) e^{-i\omega t}. \quad (\text{C.9})$$

If we now Fourier transform both sides of this equation, we see that

$$U_{fl}(\mathbf{r}_s, \mathbf{r}, \mathbf{r}_d, \tau, \omega) = U(\mathbf{r}_s, \mathbf{r}, \omega) \frac{\eta}{1 - i\omega\tau} U'(\mathbf{r}_d, \mathbf{r}, \omega). \quad (\text{C.10})$$

This formula is for a point fluorophore. We must integrate over all fluorophores to model a distribution,

$$U_{fl}(\mathbf{r}_s, \mathbf{r}_d, \tau, \omega) = \int_{-\infty}^{\infty} d\mathbf{r} U(\mathbf{r}_s, \mathbf{r}, \omega) \frac{\eta}{1 - i\omega\tau} U'(\mathbf{r}_d, \mathbf{r}, \omega). \quad (\text{C.11})$$

This solution agrees with formulations put forth by other researchers [68, 83]. It is clear that the Fourier domain equation is much less computationally intensive than the double convolution in the time domain equation.

## Appendix D

### Back Projection

We have begun to investigate a simple back projection routine in conjunction with NIM Inc., Philadelphia, PA. At NIM, Robert Danen, Yung Weng and William Thayer are currently building a frequency domain instrument which is capable of the following:

- The source modulation is variable from 10 MHz to 500 MHz.
- APD detection.
- Source fiber multiplexor allows up to 16 source position without repositioning the source fibers.

The goal of the project is to build a device capable of creating a real-time spectroscopic image of a baby brain using a hand-held optical probe. The probe, which houses 9 source fibers and 4 detector fibers, is shown in figure D.1.

Using this instrument, we measure the amplitude and phase for each source-detector pair on a sample with known optical properties. We use these measurements and the theory of DPDW's in homogeneous, semi-infinite media to calibrate the amplitude and phase of each source-detector pair. Using a single pad, with 9 sources and 4 detectors, this provides 36 measurements of amplitude and phase.

Next the pad is moved to the sample we wish to image. Again we measure the amplitude and phase for each source-detector pair. Using the calibration measurements and the homogeneous diffusion equation, we can easily solve for the absorption and scattering coefficient of the medium. As we discussed earlier, the values we obtain for the absorption and scattering coefficient represent average values.

In this work we have concentrated solely on absorption imaging. To obtain an image from the average absorption values, we must back project the calculated absorption values into the medium. This means that we have a measurement, and we are trying to guess how much of this measurement can be attributed to each volume element in the medium. For example, a particular measurement is unlikely to be affected by a volume element far away from the source and detector. It is most likely to be affected by the volume elements closest to the source or detector.

To figure out the importance of each voxel, we use a simple probability analysis. Consider a volume element,  $v$ , centered at a position  $\mathbf{r}$ . We assume that the amplitude of the DPDW which reaches  $\mathbf{r}$  is proportional to the probability that a photon from the source would reach  $\mathbf{r}$ . Based on the theory of DPDW's in a semi-infinite medium, we can calculate the amplitude of the DPDW. Similarly we can calculate the amplitude of the DPDW which travels from a source at position  $\mathbf{r}$  to the detector. We use the product of these two amplitudes for our weights ( $W$ ), and using the weights, we back project the calculated average absorption as such;

$$\mu_j = \frac{\sum_i W_{ij} \bar{\mu}_{ai}}{\sum_i W_{ij}}, \quad (\text{D.1})$$

where  $j$  enumerates the voxels and  $i$  enumerates the various measurements. Using this algorithm, we were able to create the images shown in figure D.2. The sample contains a single perfectly absorbing heterogeneity. 36 measurements of amplitude and phase at 190 MHz were used to generate the image. Note that the image does not obtain the correct object position. This is partially because the sample we measured has multiple boundaries which we have not taken into account.

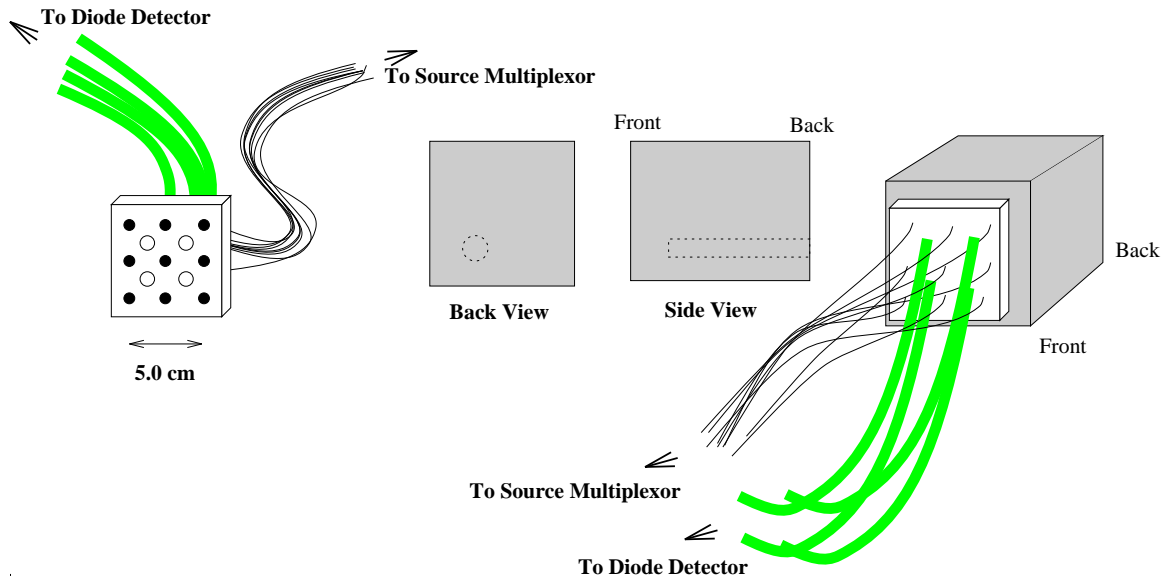


Figure D.1: The measurement geometry for the NIM back-projection imaging project.

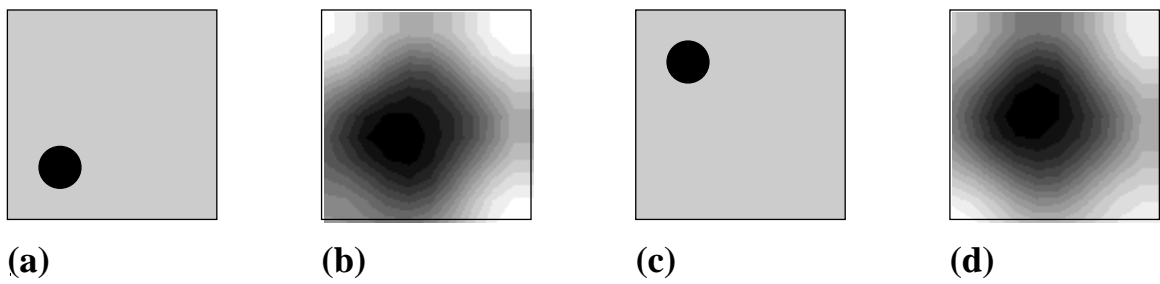


Figure D.2: Sample back-projection images for a solid block sample with an embedded heterogeneity.



## Appendix E

### The Photon Migration Imaging Software Package

PMI is a general purpose program (developed by David Boas, Maureen O’Leary, and Xingde Li at the University of Pennsylvania, and Martin Ostermeyer at the University of Texas/ M.D. Anderson Cancer Center) for solving various forward and inverse problems related to photon migration. The purpose of the program is to enable researchers to explore the propagation of diffusing photons through general heterogeneous, turbid media and to derive information about such turbid media from measurements of the diffusing photons. PMI has a modular design to facilitate the addition of new algorithms as they become available. The following modules are currently available:

- **EXACT:** Various analytic solutions to the diffusion equation are used for calculating the scattering of Diffuse Photon Density Waves (DPDW’s) from spherical and cylindrical objects embedded in infinite or semi-infinite media as well as the propagation of DPDW’s in homogeneous media.
- **TIME DOMAIN:** This module enables calculations to generate data in the time domain and enables time domain data to be used for the inverse problem. The forward problem is done by doing the calculations in the frequency domain and FFT’ed to the time domain. Experimental data in the time domain can be loaded. It is then FFT’ed to the frequency domain for analysis.
- **LAYERED:** This module provides an analytic solution for a two layered, semi-infinite system. The source and detector are assumed to be on the air-top layer interface. The bottom layer is displaced a user definable distance from the air-top

layer interface.

- **FLUORESCENCE:** This module contains analytic solutions for the generation and propagation of fluorescent DPDW's in media with spatially uniform and spatially varying optical properties.

- **FIT:** This is a chi-squared squares fitting algorithm for analyzing experimental or simulated data to find the optical properties of the background media, optical properties of any spherical or cylindrical objects, and/or the position of any spherical or cylindrical objects.

- **RECONSTRUCTION:** This program uses an iterative routine (SIRT) to solve for the optical properties of an inhomogeneous medium. The inversion will reconstruct absorption and/or scattering, fluorescence lifetime, Brownian diffusion, or shear flow. The theory is based on the First Born or First Rytov approximation and forward data may be generated using these algorithms.

- **MIE:** This module provides commands for calculating the scattering coefficient and anisotropy factor of spherical dielectrics using Mie Theory. The wavelength dependent index of refraction for water and polystyrene is included in this module.

The input is line oriented and can come either from an input file or from the standard input (keyboard). Normally the input file is specified in the command line when the program is started:

```
unixprompt> pmi [inputfilename]
```

The program will execute the commands in the input file and return to the Unix prompt. If no input file is specified, the program switches to the interactive mode and waits at its prompt for input. Now command after command can be typed in, or an inputfile can be specified with the Include command:

```
|PMI> Include [input file name]
```

After the commands in the inputfile are executed, the program will return to its prompt and wait for more commands. Any text after a # sign is taken to be a comment.



Sample input files (scripts) are shown below. Note that the imaging script uses the main shell to set up the medium, etc., the exact module to calculate the forward data, and the reconstruction module to invert the data.

## E.1 Sample PMI scripts

Sample scripts to run with PMI VERSION 2.0

List of sample scripts:

- \* exact1.pmi - Calculates the DPDW distorted by a spherical absorber.
- \* fit1.pmi - Fits data for a distorted DPDW for the position and radius of the spherical absorber.
- \* image1.pmi - Creates an image of the absorption using a reconstruction technique.

----- EXACT1.PMI -----

Following is a sample script to calculate the DPDW distorted by a spherical absorber. A 200 MHz source is positioned at the origin with an initial amplitude of 1 and phase of 0. The absorber has a radius of 0.5 cm and is centered at  $x=0$ ,  $y=2$ , and  $z=0$ . The detector is scanned from  $x=-4$  cm to  $x=4$  cm at  $y=4$  cm and  $z=0$  cm.

```
# The following script will use the analytic solution for the
# scattering of DPDW's from spherical objects to calculate the
# amplitude and phase of the distorted DPDW.
```

```
# Define the medium used for the background
#      [name] [mua] [mus] [g] [n]   [zeros]
Medium Tissue .1   10   0   1.333 0 0 0
Background Tissue
```

```
# Define the medium for the spherical inhomogeneity
#      [name] [mua] [mus] [g] [n]   [zeros]
Medium Tumor .5   10   0   1.333 0 0 0
```

```

# Define the sphere
#           [obj_name] [medium_name] [x] [y] [z] [r]
ObjectSphere Tumor          Tumor    0  2  0  .5

# Create the source-detector list
CreateSDList grid 1 0 0 0 1 0 200 200 1 0 0 1 -4 4.1 1 4 4 1 0 0 1

# RUN EXACT
Exact

#Save the source-detector list
#           [filename]
SaveSDList foo

#Save the data list
#           [FormatType] [DataType] [filename]
SaveDataList NORMAL          NORMAL    foo

```

```
----- fit1.pmi -----
```

Fits the data created with exact1.pmi for the x position and radius of the absorbing sphere.

```

# This script includes the script exact1.pmi to calculate the
# distorted DPDW and then uses the FIT module to fit for the x
# position and radius of the sphere.

```

```

# include the exact1.pmi script
Include exact1.pmi

```

```

# set the unknown parameters
SetFitParam xo .1 .5
SetFitParam a 1 .5

```

```

# Fit for the unknown parameters
Fit

```

```

# Save the result
SaveFitResult foo DATA PARAM NOAPPEND

```

```
----- image1.pmi -----
```

Defines a set of measurements, an spherical inhomogeneity, and the data that would be measured. Then inverts this data to solve for the absorption as a function of position.

```
# this script should produce a nice image of a sphere with a max
# absorption of .00303

#          mua mus g  n   db
DefineMedium back .03 10. 0. 1.333
DefineMedium obj  .04 10. 0. 1.333

# SET UP THE GEOMETRY
background back
geometry infinite
RegionBox entire back -3.0 -3.0 -.0 3.0 3.0 .0

#for reconstruction: solve for absorption, use both amp & phase data,
#          use the born approx, 1000 SIRT iterations, use the
#          homogeneous Green functions
SetReconParams ABSORPTION BOTH BORN 1000 DPDW

# constrain the absorption change to be positive
SirtConstraints 0 50 0 50 0 50

# place a sphere in the medium
objectsphere test1 obj .5 -.5 0. .5

# scan around the sides of a box
CreateSDList tandem 1 3 0 0 1 0 200 200 5 0 0 .5 -3 3 .5 0 0 .5 -3 0 0
CreateSDList tandem 1 0 3 0 1 0 200 200 5 -3 3 .5 0 0 .5 0 0 .5 0 -3 0

# use the exact module to create the forward data
Exact

# subtracts the incident from the total wave - no noise added
BornData 0. 0.

# free the object list to start with a guess of zero
freeobjectlist

# set up the region to reconstruct
Initimage -4.0 4.0 .25 -4.0 4.0 .25 -.0 .0 .58
```

```
#create the weights
createweights

# do the first order reconstruction
runsirt
saveimage filenm
```

## E.2 PMI Command Summary

PMI Online Help from 10 July 96

Version 2.1

---

### PMI Command Groups

- \* General: General commands
- \* Programming: Script programming commands
- \* Sources, Detectors, and Data: Commands for manipulating sets of sources, detectors, and data.
- \* Medium: Commands for defining the properties of different turbid media.
- \* Boundaries: Commands for defining the boundaries of a system and the boundary condition.
- \* Objects: Commands for defining inhomogeneities
- \* Images: Commands for manipulating Images of the optical properties and the fluence.

### Commands associated with different modules

- \* Exact Module: Commands related to the EXACT module. A description of the EXACT module is found [here](#).
- \* Time Domain Module: Commands related to performing calculations in the time domain.
- \* Layered Media Module: Commands related to the layered media module.
- \* Fluorescence Module: Commands related to the Fluorescence module.
- \* Fit Module: Commands related to running the FIT module.
- \* Reconstruction Module: Commands for reconstructing optical properties from measurements of the fluence. A description of the RECONSTRUCTION module is found [here](#). Refer to the sample script `image1.pmi` for an example.

- \* MIE Module: Commands for calculation of the scattering coefficient and anisotropy factor of spherical dielectrics based on Mie Theory. The wavelength dependent index of refraction for water and polystyrene is included in this module.

-----

#### PMI Command Summary

- \* General: General commands
  - o Include, Prompt, Interactive, System, Bell, Echo, Exit, Quit, Help, SetHTMLhelp, SetPMIdir, Pipes
- \* Programming: Script programming commands
  - o For...Next Loops, Variables, Operations, While Blocks, If Blocks, Functions, SetVariableVormat, Input, PassedArguments, StringCompare
- \* Sources, Detectors, and Data: Commands for manipulating sets of sources, detectors, and data.
  - o For automatic creation of whole SDLists: CreateSDLList, SaveSDLList, SaveDataList LoadSDDLList, LoadSDLList, FreeSourceList, AddNoise, AddConstantNoise Seed
  - o For manual creation of SDDSets: OpenNewSDDSSet, CloseNewSDDSSet, AddSingleSource, AddDetectorLine, AddDetectorPlane, AddDetectorVolume, PrintDetector
- \* Medium: Commands for defining the properties of different turbid media.
  - o Medium, DefineMedium, DefineMediumDynamics, DefineMediumFlourescence, PhotonDiffusion, Background, ShowMedium, FreeMedium, RegionSphere, RegionCylinder, RegionBox
- \* Boundaries: Commands for defining the boundaries of a system and the boundary condition.
  - o Plane, BoxPlanes, FreePlane, BoundaryCondition, Geometry
- \* Objects: Commands for defining inhomogeneities
  - o ObjectSphere, ObjectCylinder, ObjectBox, ObjectGauss, ApplyObjects, ShowObject, FreeObjectList
- \* Images: Commands for manipulating Images of the optical properties and the fluence.
  - o InitImage, SaveImage, SaveDynamicImage LoadImage ,

#### Commands associated with different modules

- \* Exact Module: Commands related to the EXACT module. A description

- of the EXACT module is found here.
- o Exact, Moments, Smallest
  - \* Time Domain Module: Commands related to performing calculations in the time domain.
    - o Freq2Time, SaveTimeData, SaveTimeData\_WWWStandard, TRS2TDD, LoadTimeData
  - \* Layered Media Module: Commands related to the layered media module.
    - o Layered
  - \* Fluorescence Module: Commands related to the Fluorescence module.
    - o FLR\_Exact
  - \* Fit Module: Commands related to running the FIT module.
    - o Fit, SaveFitResult, SetFitParam, SetFitTolerance, SetFitMaxIter
  - \* Reconstruction Module: Commands for reconstructing optical properties from measurements of the fluence. A description of the RECONSTRUCTION module is found here. Refer to the sample script `image1.pmi` for an example.
    - o SetReconParams, NoSirtConstraints, PertData RytovData , BornData, CreateWeights, SaveWeights, RunSirt, SirtConstraints, NoSirtConstraints
  - \* MIE Module: Commands for calculation of the scattering coefficient and anisotropy factor of spherical dielectrics based on Mie Theory. The wavelength dependent index of refraction for water and polystyrene is included in this module.
    - o Lambda, Mie, MiePoly

Examples can be found in the sample scripts. Check also the PMI homepage.

---

#### PMI Command Index

- \* AddConstantNoise
- \* AddDetectorLine
- \* AddDetectorPlane
- \* AddDetectorVolume
- \* AddNoise
- \* AddSingleSource
- \* ApplyObjects
- \* Background
- \* Bell

- \* BornData
- \* BoundaryCondition
- \* BoxPlanes
- \* CloseNewSDDSet
- \* CreateSDList
- \* CreateWeights
- \* DefineMedium
- \* DefineMediumDynamics
- \* DefineMediumFlourescence
- \* Echo
- \* Exact
- \* Exit
- \* Fit
- \* SetFitTolerance
- \* FLR\_Exact
- \* For...Next Loops
- \* FreeMedium
- \* FreeObjectList
- \* FreePlane
- \* FreeSourceList
- \* Freq2Time
- \* Functions
- \* Geometry
- \* Help
- \* If Blocks
- \* Include
- \* InitImage
- \* Input
- \* Interactive
- \* Lambda
- \* Layered
- \* LoadSDDLlist
- \* LoadSDList
- \* LoadTimeData
- \* Medium
- \* Mie
- \* MiePoly
- \* Moments
- \* NoSirtConstraints
- \* ObjectBox
- \* ObjectCylinder
- \* ObjectGauss
- \* ObjectSphere

- \* OpenNewSDDSet
- \* Operations
- \* PassedArguments
- \* PertData
- \* PhotonDiffusion
- \* Pipes
- \* Plane
- \* PrintDetector
- \* Prompt
- \* Quit
- \* RegionBox
- \* RegionCylinder
- \* RegionSphere
- \* RunSirt
- \* RytovData
- \* SaveDataList
- \* SaveDynamicImage
- \* SaveFitResult
- \* SaveImage
- \* SaveSDList
- \* SaveTimeData
- \* SaveTimeData\_WWWstandard
- \* SaveWeights
- \* Seed
- \* SetFitMaxIter
- \* SetFitParam
- \* SetHTMLhelp
- \* SetPMIdir
- \* SetReconParams
- \* SetVariableFormat
- \* ShowMedium
- \* ShowObject
- \* SirtConstraints
- \* Smallest
- \* StringCompare
- \* System
- \* TRS2TDD
- \* Variables
- \* While Blocks
- \* LoadImage

---



## Command Descriptions

-----  
General Commands

Following is a list of miscellaneous commands for PMI:

- \* Quit or Exit - to exit PMI.
- \* Include [filename] - Used to execute the commands in the script given by [filename]. Included files may in turn use the Include command. If [filename] could not be found it is searched for in the PMI directory.
- \* Interactive - switches the program to the interactive mode. This makes only sense as a command in an input file. A single point in a line or ^d (Control d) terminates the interactive mode, the control of the program returns to the superior file from which the Interactive command was invoked. If the interactive mode was already the top control level, the program terminates.
- \* System [unix command] or: ! [unix command] - launches a unix shell command.
- \* Help [command] or ? [command]- gives help for the specified [command] or general help if [command] is missing. Gives HTML-help if a browser and the help file have been specified properly with the command SetHTMLhelp, or plain text help at the PMI-prompt otherwise.
- \* SetPMIdir [pmi\_dir] - Sets [pmi\_dir] as the PMI directory. It is convenient to use this command in the initialization file ".pmi\_init" that is executed at startup.
- \* SetHTMLhelp [browser] [file] - Sets the paths for a HTML-browser (i.e. Mosaic or Netscape) which is used to display the HTML-help-file (normally "[pmi\_dir]/Help/PMI\_HELP.html"). It is convenient to use this command in the initialization file ".pmi\_init" that is executed at startup.
- \* Bell - Rings a bell.
- \* Echo [text] - writes text on the screen and to the log file
- \* Prompt - The program waits for a "RETURN". If one types "i" "RETURN", the program executes the Interactive command.
- \* Pipes - Pipes are used to control command driven external programs i.e. plot programs like gnuplot.
  - o OpenPipe [id] [prog\_name] - Start
    - + [id] - Identifies the pipe
    - + [prog\_name] - Name (with path) of the program to be

executed

- o ToPipe [id] [command] - Sends [command] to pipe [id]
- o ClosePipe [id] - Closes pipe [id]
- o CloseAllPipes - Closes all pipes. Normally obsolete since all pipes are closed automatically when the program terminates.

---

## Programming Commands

Following is a list of programming commands for use in scripts:

- \* For... Next Loops - For Next loops can be employed in scripts to repeat a block of commands with an incrementing variable. Up to 9 nested loops can be used.
  - o For [Variable Name] = [min] to [max] step [step] - Start of the For Next loop
    - + [Variable Name] - Float variable to increment.
    - + [min] - Start the loop with a value of [min]
    - + [max] - End the loop with the value of [max]
    - + [step] - increment the value by [step]
  - o Next - marks the end of the For Next loop

- \* Variables There are two types of variables: float variables and string variables. The name of a float variable is preceded by an & and the name of a string variable is preceded by an \$.

Following are examples of assignments:  
 10. o &musp = 10 Which gives the float variable &musp the value of

- o &\$filename = foobar Which gives the string variable \$filename the value of foobar.

The values of these variables are used in other commands as follows

- o DefineMedium tumor 0.1 &&musp 0 1.333
- o SaveDataList Normal Normal \$\$filename NoAppend

The format used for printing the values of these variables is set using the SetVariableFormat command.

- \* Operations Mathematical operations are available. They are addition(+), subtraction(-), multiplication(\*), and division(/). Also available are trigonometric functions (sin, cos, tan, asin, acos, atan) and exponentials (exp). These

operations are used in variable assignments. For example:

- o `&musp = &&musp + 5`
- o `&D = 2.25e10 / 3 / &&mutr`
- o `&x = 3. * Cos 30 + 1.`
- o `$filename = data_ + &&omega + .dat`

NOTE that operations are performed from left to right. For string variables, only the addition operation is available. Arguments for (sin, cos, tan) are in degrees. Arguments returned by (asin, acos, atan) are in degrees.

- \* While Blocks - While blocks are used as conditional loops in scripts. There can be 9 nested while blocks.
  - o While [value1] [condition] [value2] - Start of the While Block.
    - + [value1] and [value2] are the values being compared.
    - + [condition] - Six conditions are supported: less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), equal to (=), and not equal to (!=).
  - o Wend - End the While Block.
- \* IF Blocks - If blocks are used to execute a set of commands depending on the result of a condition. There is no limit to the number of nested If Blocks.
  - o If [value1] [condition] [value2] - Start of the If Block.
    - + [value1] and [value2] are the values being compared.
    - + [condition] - Five conditions are supported: less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), equal to (=), and not equal to (!=).
  - o IfEnd - End the If Block.
- \* Functions - Certain commands are capable of returning values to variables in the script, e.g. FIT. The usage is:
  - o COMMAND [parameters] -> &1 &2... The -> indicates that the COMMAND will return values to the variables &1 &2.... The number of values returned depends on the COMMAND.
- \* SetVariableFormat [name] [format] - Used to set the format for printing string and float variables.
  - o [name] is the name of a string or float variable. Remember to precede the string (float) variable with a \$ (&).
  - o [format] is the format string. Standard-C format strings are used. Therefore, string variables must have a string format and float variables must have a float format. %s and %f are the default formats for string and float variables respectively.

- \* Input [variable\_name] [text string] - Used to get input from the user and place the result in a variable. For example,
  - o Input &nbsp;What is the reduced scattering coef.?
- \* PassedArguments [var1] [var2]... - Used if arguments are passed to an included file. There are two ways to pass arguments to an include file: from the unix prompt you follow the format
  - o unixprompt> pmi [include\_file] [var1] [var2] ... or from the PMI interactive mode
    - o |PMI> Include [include\_file] [var1] [var2] ... The arguments are then passed to variables in the include file by using this command in the include file. PassedArguments can be used anywhere in the include file and can be used any number of times. For example, if you have an include file that expects musp and mua to be passed as arguments, then include the following line at the top of the include file:
      - o PassedArguments &nbsp; The float variable &nbsp; is set equal to the number of passed arguments when a new script is included.
- \* StringCompare [var1] = [var2] - This FUNCTION does a string compare. It returns 0 if the strings are equal and 1 if they are not equal. The compare is not case sensitive. Example:
  - o StringCompare \$\$FitWhat = MUSP -> &nbsp;

---

## Object Commands

The following commands are used for defining different inhomogeneities:

- \* ObjectSphere [name] [medium] [x] [y] [z] [r] - Used to define a spherical inhomogeneity.
  - o [name] - is the name of the sphere. The name may be exploited in future versions of PMI.
  - o [medium] - is the name of the medium which fills the sphere, e.g. tumor or hematoma. The medium must be defined prior to execution of ObjectSphere using the command DefineMedium
  - o [x] [y] [z] - define the position of the center of the sphere with units of cm.
  - o [r] - is the radius of the sphere in units of cm.
- \* ObjectCylinder [name] [medium] [x1] [y1] [z1] [x2] [y2] [z2] [r] - Used to define a cylindrical inhomogeneity.
  - o [name] - is the name of the cylinder. The name may be

- exploited in future versions of PMI.
- o [medium] - is the name of the medium, see in ObjectSphere.
- o [x1] [y1] [z1] - defines the position of the center of the bottom of the cylinder (units of cm).
- o [x2] [y2] [z2] - defines the position of the center of the top of the cylinder (units of cm). Note that EXACT can only work with infinite cylinders and thus only uses these positions to define the axis of the cylinder. Presently, EXACT can only work with cylinders which have an axis parallel to the z-axis.
- o [r] - is the radius of the cylinder (units of cm).
- \* ObjectBox [name] [medium] [xlo] [ylo] [zlo] [xhi] [yhi] [zhi] - Used to define a rectangular inhomogeneity with sides parallel to the coordinate axes.
  - o [name] - is the name of the box. The name may be exploited in future versions of PMI.
  - o [medium] - is the name of the medium, see in ObjectSphere.
  - o [xlo] [ylo] [zlo] - define the lower x, y, z values (units of cm).
  - o [xhi] [yhi] [zhi] - define the upper x, y, z values (units of cm).

---

### Medium Commands

The following commands are used to define the properties of different media:

- \* Medium - THIS COMMAND IS OBSOLETE. Use DefineMedium instead.
- \* DefineMedium [name] [mua] [mus] [g] [n] - Used to define or change the properties of a medium.
  - o [name] - is the name of the medium which is used by other parts of the program to reference a particular medium.
  - o [mua] - is the absorption coefficient of the medium.
  - o [mus] - is the scattering coefficient of the medium.
  - o [g] - is the average cosine of the scattering angle.
  - o [n] - is the index of refraction.
- \* DefineMediumDynamics [name] [Brownian] [Shear] - Used to define or change the DYNAMICAL properties of a medium.
  - o [name] - is the name of the medium which is used by other parts of the program to reference a particular medium.
  - o [Brownian] - is the Brownian diffusion coefficient of the scattering particles in units of  $\text{cm}^2/\text{s}$ .

- o [Shear] - is the rate of shear for the scattering particles in units of 1/s.
- \* DefineMediumFluorescence [name] [tau] [mua\_f] [musp\_f] [fmua] [fmusp] [fmua\_f] [fmusp\_f] - Used to define or change the FLUORESCENT properties of a medium.
  - o [name] - is the name of the medium which is used by other parts of the program to reference a particular medium.
  - o [tau] - is the fluorescence lifetime.
  - o [mua\_f] - is the absorption coefficient of the chromophore at the fluorescence wavelength.
  - o [musp\_f] - is the reduced scattering coefficient of the chromophore at the fluorescence wavelength.
  - o [fmua] - is the absorption coefficient of the fluorophore at the excitation wavelength.
  - o [fmusp] - is the reduced scattering coefficient of the fluorophore at the excitation wavelength.
  - o [fmua\_f] - is the absorption coefficient of the fluorophore at the fluorescence wavelength.
  - o [fmusp\_f] - is the reduced scattering coefficient of the fluorophore at the fluorescence wavelength.
- \* Background [medium\_name] - Used to set the background medium.
  - o [medium\_name] - is the name of the medium defined using the DefineMedium command.

-----

### Sources and detectors

Several sources, each with its own amplitude and phase, and several detectors, also with amplitude and phase, are organized in a Source-Detector-Data-Set (SDDS). The modulation frequency and the correlation time are the same for a SDDS and all sources are 'on' at the same time. The forward algorithms (Exact andPertData) can operate on one SDDS at a time. If one wants to calculate situations with several modulation frequencies or correlation times or with changed source or detector positions, one has to create several SDDSes. All the SDDSes are stored in a list, the SDLList.

Because inverse algorithms work with the SDLList there is a group of commands for creating, saving, and loading a whole SDLList as well as saving and loading the corresponding data in one step. For pure forward calculations it can be more convenient to set a SDDS 'by hand', which is done by the second group of commands.

### Commands for SDLLists

- \* CreateSDList [format] [parameters...] - Used to create a source-detector set.
  - o [format]=GRID This format is used to create a set of detectors for a spatially fixed source group. The modulation frequency and correlation time of the source can be scanned as well as the X, Y, and Z coordinates of the detector.
    - + [parameters...] =
    - + [number of Sources in Source Group]
    - + [Xs] [Ys] [Zs] [Amp] [Ph] ... Position and initial amplitude and phase of each source in the source group.
    - + [Omega Min] [Omega Max] [Omega Step] Frequency given in Mhz
    - + [Tau Min] [Tau Max] [Tau Step] Correlation time in seconds
    - + [X Min] [X Max] [X Step] X coordinates of the detector
    - + [Y Min] [Y Max] [Y Step] Y coordinates of the detector
    - + [Z Min] [Z Max] [Z Step] Z coordinates of the detector
  - o [format]=TANDEM This format is used to scan a source group and detector in tandem. A grid is defined and the position of the source group and detector are given relative to the grid points.
    - + [parameters...] =
    - + [number of Sources in Source Group]
    - + [Xs] [Ys] [Zs] [Amp] [Ph] ... Position and initial amplitude and phase of each source in the source group. The position is given relative to the grid points defined below.
    - + [Omega Min] [Omega Max] [Omega Step] Frequency given in Mhz
    - + [X Min] [X Max] [X Step] X coordinates of the grid points
    - + [Y Min] [Y Max] [Y Step] Y coordinates of the grid points
    - + [Z Min] [Z Max] [Z Step] Z coordinates of the grid points
    - + [X det] [Y det] [Z det] Position of the detector relative to the grid points
  - o [format]=CYLINDER2 This format is used to scan a source and scan a detector around a cylinder. A grid is defined and the position of the source group is given relative to the grid points. Next a detector grid is defined relative to the source grid points. Thus, for each source there are a grid of detectors.

- + [parameters...] =
- + [Zmin] [Zmax] [Zstep] The z positions for the source grid.
- + [Phis\_Min] [Phis\_Max] [Phis\_Step] The phi positions for the source grid.
- + [rho] This is the radius of the cylindrical scan.
- + [omega] Modulation frequency
- + [tau] Correlation time
- + [Num\_Srcs] number of sources in the source group
- + [Zsrc] [Phi\_src] [Amp] [Ph] ... Position and initial amplitude and phase of each source in the source group. The position is given relative to the source grid points.
- + [Z\_det] [Phi\_Min\_Det] [Phi\_Max\_Det] [Phi\_Step\_Det] Definition of the detector grid relative to the source grid.
- o [format]=TDTANDEM This format is used to define a pulsed source. The source and detector are scanned in tandem on a grid. The source and detector positions are defined relative to the grid.
  - + [parameters...] =
  - + [xmin] - define the x part of the grid
  - + [xmax]
  - + [xstep]
  - + [ymin] - define the y part of the grid
  - + [ymax]
  - + [ystep]
  - + [zmin] - define the z part of the grid
  - + [zmax]
  - + [zstep]
  - + [xsource]- Separation between grid point and source
  - + [ysource]
  - + [zsource]
  - + [amp] - Initial amplitude of source
  - + [ph] - Initial phase of source
  - + [xdetectr] - Separation between grid point and detector
  - + [ydetector]
  - + [zdetector]
  - + [tmax] - Amount of time to watch the response to a pulse at time = 0. Units of seconds. This sets the frequency step, i.e. freq. step = 1/[tmax].
  - + [number bins] - Number of bins to divide [tmax] in



- to. Must be a power of 2.
- + [freq max] - Maximum frequency for calculations. Units of MHz. If this is too small then oscillations will appear in the time domain data as artifacts.
- \* SaveSDLList [filename] - Used to save a list of source-detector sets
  - o [filename] - is the filename for the output file. Note that .sd is appended to the filename
- \* SaveDataList [FormatType] [DataType] [filename] [append] - Used to save a list of data.
  - o [FormatType] One of the following:
    - + Normal - creates a data file with the standard PMI code format. [DataType] can have the following values:
      - + Normal - saves the incident and total amplitude and phase.
      - + Total/Incident - saves the ratio of the total to the incident.
    - + WWWstandard - uses the format designed for sharing data over the WWW. [DataType] can have the following values:
      - + Normal - saves the incident and total amplitude and phase.
  - o [DataType] - determines what type of data is saved, e.g. save the total amplitude and phase or relative to the incident amplitude and phase. The value of this parameter depends on the chosen [FormatType].
  - o [filename] - is the filename for the output file. Note that .dat is appended to the filename
  - o [append] - if this equals APPEND then the output is appended to [filename]
- \* AddNoise [shot noise] [position uncertainty] - Used to add noise to a set of data.
  - o [shot noise] - When this is not equal to zero then shot noise is added to the data, The noise is determined by taking the square-root of the fluence multiplied by [shot noise]. [shot noise] is thus the product of the source amplitude in units of photons per second, the signal integration time in units of seconds, the detector area in units of  $\text{cm}^2$ , and the quantum efficiency of the detector.
  - o [position uncertainty] - When this is not equal to zero then random noise is added to the data as given by the uncertainty of the position of the source and/or detector relative to each other and/or relative to the sample. [position uncertainty] is given in units of cm.

- \* AddConstantNoise [amp error] [phase error] - Used to add CONSTANT noise to a set of data. [amp error] is given as a percentage. [phase error] is given in degrees.
- \* Seed [integer] - Used to seed the random number generator. Use large negative integers.

Commands for individual SDDSets

- \* OpenNewSDDSet [name] [f] [tau] - Used to create a new SDDSet with the identification [name], the modulation frequency [f] in Hz, and the correlation time [tau] in seconds. After this command sources and detectors can be added until the SDDSet is closed with CloseNewSDDSet.
- \* CloseNewSDDSet [name] - Used to close the SDDSet with the name [name].
- \* AddSingleSource [amplitude] [phase] [x] [y] [z] - Adds a source to the currently open SDDSet.
  - o [amplitude] in W
  - o [phase] in degree
  - o [x] [y] [z] position in cm
- \* AddDetectorLine [name] [amp] [ph] [x] [y] [z] [e1x] [e1y] [e1z] [lo1] [hi1] [step1] - Adds a line of detectors with arbitrary orientation to the currently open SDDSet.
  - o [amp] [ph] amplitude and phase for the whole line.
  - o [x] [y] [z] position
  - o [e1x] [e1y] [e1z] vector that specifies the direction of the line, starting from (x,y,z).
  - o [lo1] [hi1] [step1] specifies distribution of detectors along the line. The zero of the coordinate system is at (x,y,z), the direction is specified by the vector e1.
- \* AddDetectorPlane [name] [amp] [ph] [x] [y] [z] [e1x] [e1y] [e1z] [lo1] [hi1] [step1] [e2x] [e2y] [e2z] [lo2] [hi2] [step2] - Adds a plane of detectors with arbitrary orientation to the currently open SDDSet.
  - o [amp] [ph] amplitude and phase for the whole plane.
  - o [x] [y] [z] position of the origin of the new coordinate system.
  - o [e1x] [e1y] [e1z] and [e2x] [e2y] [e2z] are two vectors e1 and e2 that specify the axes of the new coordinate system. The zero of the new coordinate system is at (x,y,z).
  - o [lo1] [hi1] [step1] and [lo2] [hi2] [step2] specify the sampling along the axes of the coordinate system. The zero of the coordinate system .
- \* AddDetectorVolume [name] [amp] [ph] [x] [y] [z] [e1x] [e1y] [e1z] [lo1] [hi1] [step1] [e2x] [e2y] [e2z] [lo2] [hi2] [step2] [e3x]

[e3y] [e3z] [lo3] [hi3] [step3] - Adds a rectangular volume of detectors with arbitrary orientation to the currently open SDDSet. Syntax similar to AddDetectorPlane but with an additional vector e3.

- \* PrintDetector [name] [printwhat] [style] [filename] - Writes values at detectors.
  - o [name] name of a detector that was defined with AddDetectorLine or AddDetectorPlane.
  - o [printwhat] ... sorry, no complete description yet, some of many choices are:
    - + AM\_P - amplitude of perturbation fluence
    - + PH\_P - phase of perturbation fluence
    - + AM\_F - amplitude of total fluence
    - + PH\_F - phase of total fluence
    - + AM\_B - amplitude of background fluence
    - + PH\_B - phase of background fluence
    - + PH\_DIF - phase difference between total and background fluence
    - + AM\_RAT - amplitude ratio between total and background fluence
  - o [style] can be one of the following:
    - + GNU - file can be plotted directly with gnuplot.
    - + MATH - file can be plotted directly with Mathematica. In Mathematica type: <<filename
  - o [filename] specifies the file. If no filename is given, PrintDetector writes on the screen.

---

#### IMAGE module commands

The following commands are used for defining the image matrix of a finite system. Each voxel in the image contains the optical properties defined by the user by setting the background medium and the object list.

- \* InitImage [X min] [X max] [X step] [Y min] [Y max] [Y step] [Z min] [Z max] [Z step] - sets area to be considered. The area to be reconstructed should be a subset of this large cube. The corners on the cube and the step define the region. When InitImage is called, then optical properties of any objects which have been defined are inserted (voxel by voxel) into the image.
- \* SaveImage [filename] [parameters] - save an image to the file filename. The format of the image file is as follows:
  - o [number of x] [number of y] [number of z]

- o [xmin] [xmax] [xstep]
- o [ymin] [ymax] [ystep]
- o [zmin] [zmax] [zstep]
- o a list of the quantities that have been saved
- o the background parameters for each or these quantities

Next, the columns of data are listed. The valid parameters are scattering (MUSP), absorption (MUA), index of refraction (N), Brownian diffusion (DB), and shear (SHEAR). If NOHEADER is included in the parameter list, then the first 6 rows are omitted. Note: You cannot use LoadImage to read in a saved image if it is saved with the NOHEADER option.

- \* LoadImage [filename] - read a previously saved image into memory. The image CANNOT have been saved with the headers disabled.

---

#### BOUNDARY module commands

The following commands are used for defining the boundaries of a finite system and the boundary condition to be used in the calculations.

- \* Plane [a] [b] [c] [d] [n out] - Used to define a planar boundary.
  - o [a] [b] [c] [d] - The plane is defined by the equation  $[a] X + [b] Y + [c] Z + [d] = 0$ . Note that [a] [b] [c] is the normal vector to the plane and must point towards the turbid medium.
  - o [n out] - is the index of refraction on the free-space side of the planar boundary.
- \* BoxPlanes [x1] [y1] [x2] [y2] - Used to define a BOX geometry. The sides of the box are assumed parallel to the XZ and YZ planes. Opposite corners of the box in the XY plane are given by [x1] [y1] and [x2] [y2]. Note that [x1] [y1] must be less than [x2] [y2]. The geometry type is automatically set to BOX.
- \* BoundaryCondition [parameter] - Used to define the boundary condition. [parameter] is one of the following:
  - o Zero - uses the zero boundary condition in calculations.
  - o Extrapolated - uses the extrapolated boundary condition in calculations.
- \* Geometry [parameter] - Used to set the geometry type as given by [parameter]. [parameter] is one of the following:
  - o Infinite - Calculations are performed for an INFINITE

medium.

- o Planar - for a medium with PLANAR boundaries.
- o Box - for a medium with BOX boundaries.

-----

#### EXACT module commands

These commands will calculate the incident, scattered, and total fluence for the system defined using object, medium, and boundary commands.

- \* Exact - Used to calculate the analytic solution of the diffusion equation for the defined system.
- \* Moments [file name] [number of moments] - Used to tell EXACT how many moments to save to file name [file name] during the calculation of the scattered wave. If [number of moments] is negative then the moments are appended to the file.

-----

#### TIME-DOMAIN module commands

These commands are used to perform calculations in the time-domain. It is necessary to prepare the program for time-domain calculations either by creating pulsed sources with CreateSDLList or by loading a time-domain set of data with LoadTimeDomain.

- \* Freq2Time - All calculations are done in the frequency-domain. After the calculations, it is necessary to tell the program to Fourier transform the results into the time-domain. This is done with Freq2Time. There are no parameters. This will only work if a pulsed source has been defined with CreateSDLList or a time-domain data set was loaded with LoadTimeDomain.
- \* SaveTimeData [Format Type] [Data Type] [File name] [Append] - Used to save time-domain data to file [File name].
  - o [FormatType] One of the following:
    - + Standard - Creates a data file with the standard PMI code format. [DataType] is not used.
    - + Time - Saves the data in two column format. The first column is the time in nanoseconds and the second column is the fluence. [DataType] is not used.

- o [DataType] - Not used, however you must place a word here. Use foo if you can't think of anything else.
- o [filename] - is the filename for the output file. Note that .tdd is appended to the filename
- o [append] - if this equals APPEND then the output is appended to [filename]
- \* SaveTimeData\_WWWStandard - Don't use.
- \* TRS2TDD [File TRS] [File TDD] [nbins] [fbin] [tmax] [fmax] [xs] [ys] [zs] [xd] [yd] [zd] - Reads in one column time-domain data and saves it in the standard time-domain format. The first row of the one column data file must be the number of data points that follow in the subsequent rows.
  - o [File TRS] - This is name of the one column time-domain data file. A .trs is appended to the file name. The first row must be the number of data points that follow in the subsequent rows.
  - o [File TDD] - This is file name used to save the data in the standard .tdd format. A .tdd is appended to the file name.
  - o [nbins] - This is the number of data points to take from the .trs file and save in the .tdd file.
  - o [fbin] - This is the first data point to take from the .trs file. The next [nbins]-1 data points are saved as well. This option allows initial zeros or other data sets to be skipped.
  - o [tmax] - Since time information is not contained in the .trs file it is necessary to indicate the time range spanned by the [nbins] data points.
  - o [fmax] - This is the maximum modulation frequency calculated when the Fourier transform of this data set is taken.
  - o [xs] [ys] [zs] - Position of the source.
  - o [xd] [yd] [zd] - Position of the detector.
- \* LoadTimeData [TDD file] [reverse] [Instrument function] - This command will load time-domain data that is in the standard format and deconvolute the data if an instrument function is given.
  - o [TDD file] - This is the .tdd file. A .tdd is appended to [TDD file].
  - o [reverse] - The .tdd file must have the data in an increasing time sequence. That is the first data point is at time=0 and the last data point is at time=[tmax]. Some data files are saved in the reverse format. This is indicated by typing REVERSE here. If the data is in the proper sequence then type NORMAL.
  - o [Instrument function] - This is the file containing the

instrument function. A .inst is appended to the file name. The format is single column with the first row containing the number of data points following in the subsequent rows. The deconvolution of the data only works if the [TDD file] and [Instrument function] have the same number of data points. The time scale of the instrument function is assumed to be the same as the data.

---

#### Layered module commands

These commands are used to calculate diffuse back reflection from a layered, turbid system. Analytic solutions of the diffusion equation are used.

- \* Layered [Medium 1] [Medium 2] [d] - This is the command to start the calculation. The source-detector geometry must be defined before calling this command. The layered medium is semi-infinite with the free-space interface at  $z=0$  in the xy-plane. NOTE that the z coordinates of the sources and detectors are assumed to be zero. It does not matter if you defined them as something other than zero. The index of refraction outside of the turbid media is assumed to be 1.0.
    - o [Medium 1] is the medium for the top layer.
    - o [Medium 2] is the medium for the bottom layer.
    - o [d] - The top layer has a thickness of [d] cm.
- 

#### FLUORESCENCE module commands

These commands will calculate the incident, scattered, and total FLUORESCENT fluence for the system defined using object, medium, and boundary commands.

- \* FLR\_Exact - Used to calculate the analytic solution of the diffusion equation for the defined FLUORESCENT system.
- 

#### FIT module commands

Commands for fitting analytic solutions to experimental or simulated data follow:

- \* Fit [FitWhat] - Used to initiate the fit. The source-detector list and data must already be defined as well as all parameters which are known. Refer to the sample scripts for an example. This command is a function. It returns the chi-squared value for the fit followed by the best fit for the fit parameters. The ordering for the returned fit parameters is given by the list in SetFitParam.
  - o [FitWhat] - determines whether to fit the amplitude data (AMP), the phase data (PHASE), or both amplitude and phase (BOTH). For Example, Fit AMP or Fit PHASE or Fit BOTH
- \* SaveFitResult [filename] [DataFlag] [ParamFlag] [AppendFlag] - Saves the results from the Fit command. Typically, the results of the fit are written as the first line of the output file followed by a comparison of the "experimental" and simulated data. Each line of the data corresponds to a single source-detector pair.
  - o [filename] is the filename. An extension is added to [filename] indicating what parameters were fitted.
  - o [DataFlag] determines what type of data is saved if any. [DataFlag] can have the following parameters:
    - + NODATA - no data is saved.
    - + FREQ - each line of the data begins with the modulation freq. in MHZ of the source for that measurement.
  - o [ParamFlag] determines if the fitted parameters are saved to the file. If [ParamFlag]=NOPARAM then no parameters are saved.
  - o [AppendFlag] determines if the new data starts a new file or is appended to the end of an existing file. If [AppendFlag]=APPEND then the data is appended to the existing file.
- \* SetFitParam [parameter] [value] [uncertainty] - Used to set the initial guess and uncertainty of an unknown parameter. If you wish to make an unknown parameter known then set the [uncertainty] equal to zero.
  - o [parameter] - is one of the following
    - + Xo - x position of the object center
    - + Yo - y position of the object center
    - + Zo - z position of the object center
    - + a - radius of the object
    - + musp\_obj - scattering coefficient of the object
    - + mua\_obj - absorption coefficient of the object
    - + db\_obj - Brownian diffusion coefficient of the object
    - + musp\_back - scattering coefficient of the background
    - + mua\_back - absorption coefficient of the background



- + db\_back - Brownian diffusion coefficient of the background
- + musp\_f\_back- scattering coefficient at emission wavelength due to chromophore
- + mua\_f\_back- absorption coefficient at emission wavelength due to chromophore
- + fmusp\_f\_back- scattering coefficient at emission wavelength due to fluorophore
- + fmua\_f\_back- absorption coefficient at emission wavelength due to fluorophore
- + fmusp\_back - scattering coefficient at excitation wavelength due to fluorophore
- + fmua\_back - absorption coefficient at excitation wavelength due to fluorophore
- + tau\_f\_back - lifetime of fluorophore
- o [value] - is the initial guess for the unknown parameter
- o [uncertainty] - is an estimate of the uncertainty in the initial guess
- \* SetFitTolerance [tolerance] - Used to set the tolerance that must be met when using the FIT module to determine unknown parameters.
- \* SetFitMaxIter [max] - Used to set the maximum number of iterations for the AMOEBA fit.

---

#### Reconstruction module commands

The reconstruction module inverts the first order solution to the integral equation using either the Born or Rytov Expansion (see Kak and Slaney for descriptions of the Born and Rytov expansions for diffracting sources). The user must define the quantities to be inverted (i.e. absorption, scattering), the grid to reconstruct on, and the measurement geometry. The only inversion available here is the SIRT algorithm. Refer to the sample script image1.pmi for an example.

- \* SetReconParams [Quantity] [Data Type] [Inverse Method] [Iterations] [Forward Method] - sets up the reconstruction.
  - o [Quantity] - can be Absorption or Scattering or Both or Brownian or Shear.
  - o [Data Type] - can be Amplitude or Phase or Both.
  - o [Inverse Method] - can be Born or Rytov.
  - o [Iterations] - the number of SIRT iterations.
  - o [Forward Method] - can be DPDW to calculate the green's

function for a homogeneous medium, or exact to use the exact calculation for a sphere or cylinder, Note: exact uses the current object list to determine what homogeneities are present.

- \* RytovData [shot noise] [positional error] - divides the total measured wave by the incident and takes the log of the results. This is for the Rytov-type inversion. Shot noise and positional error are added to the measurement before the division.
  - \* BornData [shot noise] [positional error] - subtracts the incident wave from the total measured wave. This is for the Born-type inversion. Shot noise and positional error are added to the measurement before the subtraction.
  - \* CreateWeights - uses either the Born or the Rytov approximation to calculate the weights (also called photon hitting densities, or three point green's functions) to necessary for the inversion routines. WARNING: You must have a source-detector list and an image matrix to create the weights.
  - \* SaveWeights [filename] - saves the weights to a file. This routine is only recommended for debugging purposes as the weight matrix is generally quite large.
  - \* PertData [filename] - Multiplies the weights by the image to calculate first order forward data. The data is written to filename.pert.dat. If filename = none, then the data is saved in the internal source-detector-data list for further use. WARNING: the weights and the image matrix have to have been created before PertData is called.
  - \* RunSirt [flag] - starts the iterative reconstruction routine. See Kak and Slaney for details on SIRT. If the flag is set to be CALC\_WEIGHTS, then the weights are created before SIRT is started.
  - \* SirtConstraints [abs. min] [abs. max] [scat min] [scat. max] [lifetime min] [lifetime max] [Brownian min] [Brownian max] [shearmin] [shear max] - constrains the SIRT solutions for each voxel to be between some minimum and maximum values. The reconstruction quantities that may be constrained are absorption, scattering, fluorescence lifetime, Brownian diffusion, and shear flow.
  - \* NoSirtConstraints - removes any/all constraints on reconstruction quantities.
-

## MIE module Commands

This command will calculate the  $l^*$  (random walk step), which is in another word the reciprocal of the scattering coefficient  $\mu_s'$  for polystyrene spheres based on Mie Theory. It requires to get the wavelength  $\lambda$  first in the units of cm. Then either one of the following commands can be chosen depending upon the situation. A sample script can be found script mie.pmi.

- \* `Lambda [wavelength]` - Set the wavelength of light where `[wavelength]` is in units of cm.
- \* `Mie [param]` - Used to calculate the  $l^*$  (cm) given the indices of refraction of the polystyrene spheres and the solvent, the diameter of the polystyrene sphere ( $\mu\text{m}$ ), and the volume fraction of polystyrene spheres in the solvent.
- \* `MiePoly [param]` -Used to calculate the  $l^*$  (cm) if the solvent is water, given the temperature (K), the diameter of the polystyrene sphere ( $\mu\text{m}$ ), and the volume fraction of polystyrene spheres in the solvent.

Incorporating with other PMI commands, you can use the returned value ( $l^*$ ) and take advantage of the `For...Next` command. For details, please see Functions for the usage of returned values and `For...Next` for the loops.

-----

## REFERENCES

- \* D. A. Boas, M. A. O'Leary, B. Chance, and A. G. Yodh, Proc. Natl. Acad. Sci. USA, 91, 4887 (1994).
- \* A. C. Kak and M. Slaney, Principles of Computerized Tomographic Imaging, (IEEE Press, New York, 1988).
- \* M. A. O'Leary, D. A. Boas, B. Chance, and A. G. Yodh, Optics Letters, 20(5) (1995).



## Appendix F

### Parallel Processing

There are two very different methods of creating parallel code. The traditional method, employed by Cray computers, created parallel operations (addition, multiplication, etc.). Parallel operations are extremely useful in large matrix calculations. Instead of making these small operations parallel, we could make large routines parallel. In this type of parallel processing, we break up a problem into large, independent jobs, and run them each on a separate processor.

In our code we often simulate the measurements that would result from particular source and detector. Whenever we make more than one measurement, we end up with a set of source-detector pairs. These sets are stored internally as a list of simulations to do. Traditionally, the code would cycle through the list, and return a corresponding list of simulated measurements. In the new parallel code, the source-detector list is broken up, and sections of the list are sent to multiple parallel processors.

The computer system we use for parallel processing is an IBM SP2 (Scalable POWER Parallel2) system. POWER stands for performance optimized with enhanced RISC; RISC stands for reduced instruction set computer. A SP2 consists of a series of racks of nodes. Each node is a separate RISC 6000 superscalar processor. Nodes come in two flavors. The thin nodes have a 64kB cache, and the thick nodes have a 256kB cache. A rack consists of either 8 thick or 16 thin nodes. The system we used had one rack consisting of 16 thin nodes.

In order to communicate between the different nodes, we need a common language. For this work we used the IBM message passing library (MPL). Since there is no

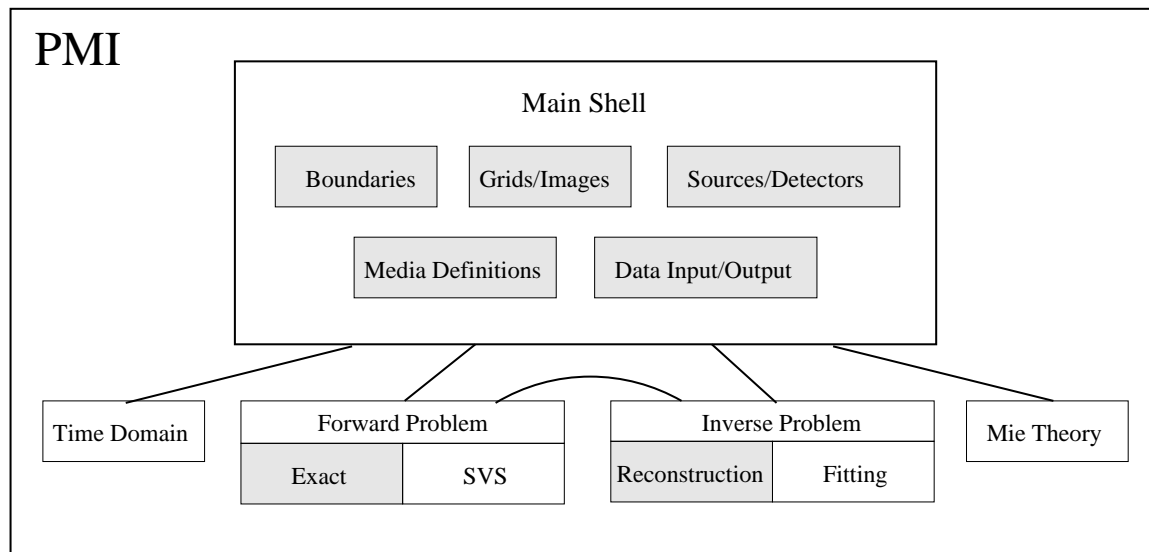


Figure F.1: An overview of the PMI software. The grey modules correspond to the modules we typically use in the tomographic reconstruction problem.

shared memory between the nodes, the MPL was used to pass all of the relevant source-detector lists, medium properties, etc.

All of the computer code used in this dissertation has been incorporated into a central package called Photon Migration Imaging (PMI). PMI is broken up into several modules as pictured in figure F.1. The routine to calculate the forward problem is called not only when we create simulated data, but also to calculate the weight for every source-voxel-detector set. (Recall that the weights are functions of the photon density at particular positions). Thus it is clear that we can greatly speed up the calculation time by making the calculation of the forward problem run in parallel.

The algorithm for parallel calculation of the forward problem is quite simple. First the code is loaded into each node. We define the 0th node as the master. The master node begins loading the medium, source and detector information as usual. The other nodes jump right to the forward calculation and wait. When the master program is given a list of source-detector pairs to calculate, it looks for a waiting node.

Suppose we have 8 nodes (7 slaves plus a master) and 8 source-detector pairs. The master node reaches a point where it is ready to calculate the forward problem. It

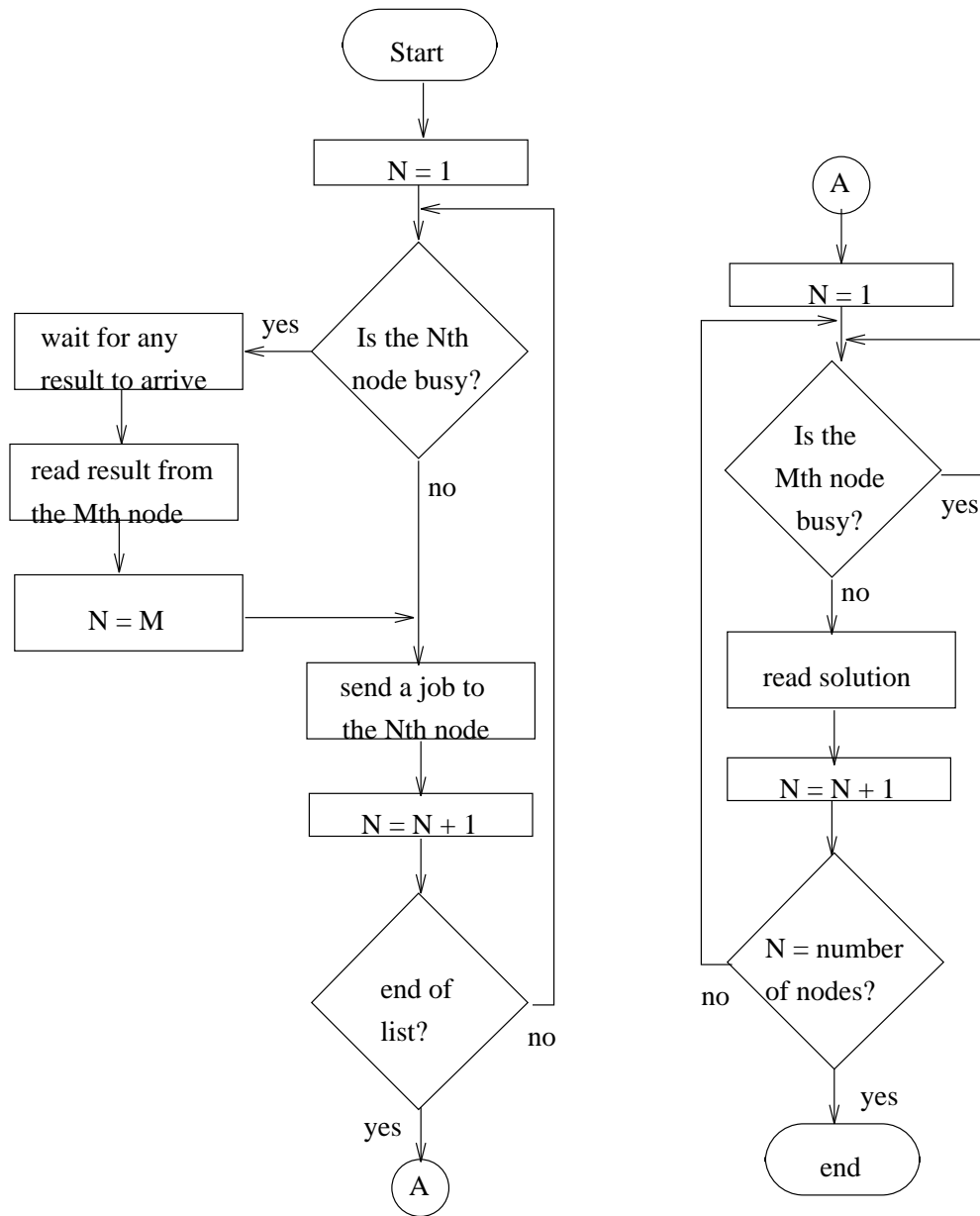


Figure F.2: A block diagram of the flow of the parallel code.

now has a list of 8 source-detector pairs. First the master node sends the first source-detector pair to node 1. The next to node 2, until all 7 available nodes are working. The master node now waits for a result to come back. When a result is returned from any of the 7 working nodes, the result is stored, and the last source-detector pair is sent to that node. When all of the source-detector pairs have been sent, the master node waits until all nodes are finished, and stores their results (see figure F.2).

Because we do not have a common, shared memory between processors, a lot of information must be passed between the master and the slave nodes. In particular, we must pass the properties of the medium, the geometry of the sample, the object information, and the source and detector information. If the calculation of the forward problem is simple, as in the case of a homogeneous, infinite medium, then this passing time is significant compared to the calculation time, and we do not expect a substantial savings from the parallel processing. However, if the forward problem is more complicated, for example a slab geometry or a heterogeneous medium, then the passing time is less important. To demonstrate this, we have run a calculation for each of the 4 sample geometries shown in the top panels of figure F.3. In each simulation, the forward problem was calculated 500 times. The results, also shown in figure F.3, clearly indicate that the parallel processing is more efficient for the more complicated system.

Finally, using a heterogeneous slab geometry (F.3d) we demonstrate the effect of adding more nodes. Table F.1 gives both the clock time and the CPU time for the same calculation done using 1, 2, 4 and 8 nodes. Using the 8 nodes, we are able to decrease the calculation time by a factor of 5.



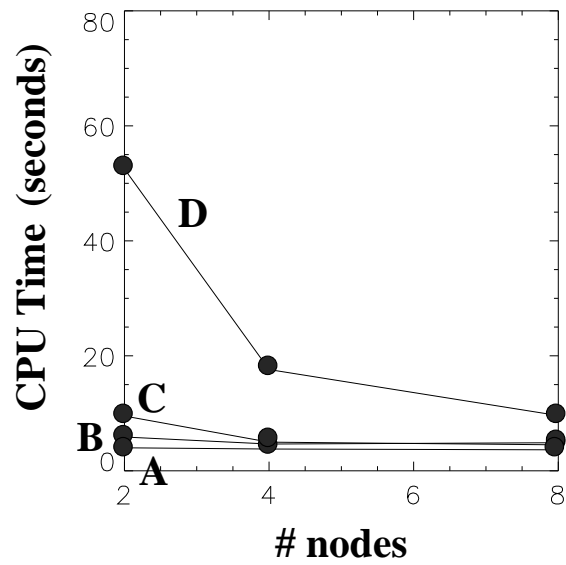
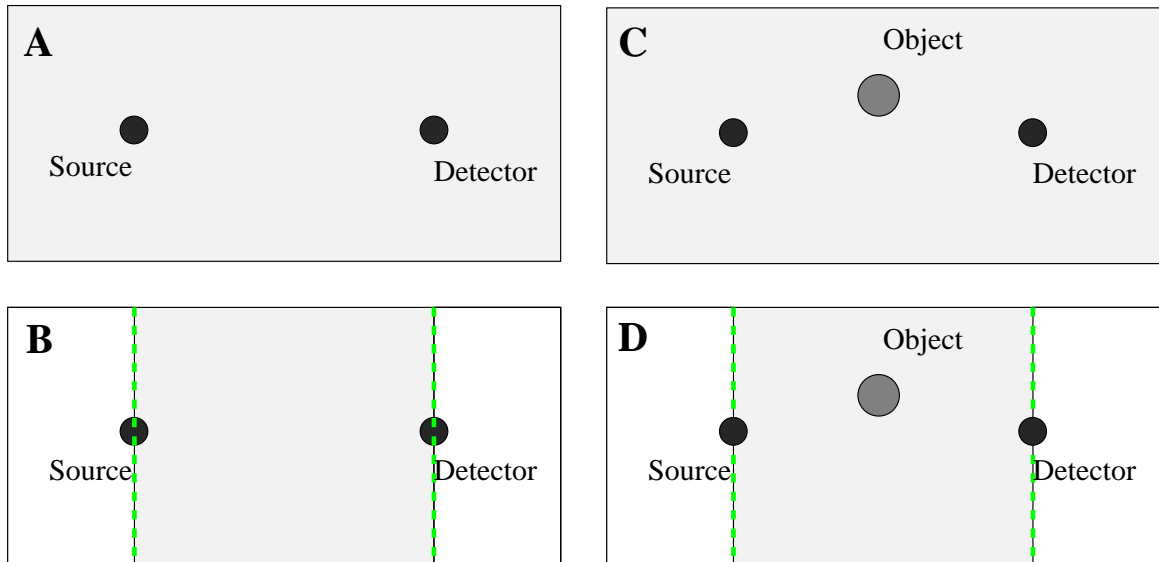


Figure F.3: The CPU times for various homogeneous and heterogeneous systems.

computer	number of nodes	Clock time (s)	CPU time (s)
SP2	1	91	44
	2	104	51
	4	38	18
	8	20	9

Table F.1: The clock and CPU times for 1, 2, 4 and 8 node trials.

## Bibliography

- [1] M. Cutler, , Surg. Gynecol. Obstet. **48**, 721 (1929).
- [2] S. Nioka, M. Miwa, S. Orel, M. Shnall, M. Haida, S. Zhao, and B. Chance, *Oxygen Transport to Tissue* (Plenum Press, New York, 1994), pp. 171–179.
- [3] M. Patterson, B. Chance, and B. Wilson, *Time resolved reflectance and transmittance for the non-invasive measurement of tissue optical properties*, Applied Optics **28(12)**, 2331 (1988).
- [4] E. Gratton, W. Mantulin, M. J. van de Ven, J. Fishkin, M. Maris, and B. Chance, *Proceeding of the Third International Conference: Peace Through Mind/Brain Science* (Hamamatsu Photonics KK, Hamamatsu, Japan, 1990), p. 183.
- [5] J. B. Fishkin and E. Gratton, *Propagation of photon density waves in strongly scattering media containing an absorbing semi-infinite plane bounded by a straight edge*, J. Opt. Soc. of America A **10**, 127 (1993).
- [6] M. A. O’Leary, D. A. Boas, B. Chance, and A. G. Yodh, *Refraction of diffuse photon density waves*, Phys. Rev. Let. **69**, 2658 (1992).
- [7] D. A. Boas, M. A. O’Leary, B. Chance, and A. G. Yodh, *Scattering and wavelength transduction of diffuse photon density waves*, Phys. Rev. E **47**, R2999 (1993).
- [8] B. J. Tromberg, L. O. Svaasand, T. T. Tsay, and R. C. Haskell, *Properties of photon density waves in multiple scattering media*, Applied Optics **32(4)**, 607 (1993).

- [9] B. Chance, K. Kang, L. He, J. Weng, and E. Sevick, *Highly sensitive object location in tissue models with linear in-phase and anti-phase multi-element optical arrays in one and two dimensions*, Proc. Natl. Acad. Sci. **90**, 3423 (1993).
- [10] D. A. Boas, M. A. O'Leary, B. Chance, and A. G. Yodh, *Scattering of diffuse photon density waves by spherical inhomogeneities within turbid media: analytic solution and applications*, Proc. Natl. Acad. Sci. USA **91**, 4887 (1994).
- [11] M. A. O'Leary, D. A. Boas, B. Chance, and A. G. Yodh, *Reradiation and imaging of diffuse photon density waves using fluorescent inhomogeneities*, Journal of Luminescence **60-61**, 281 (1994).
- [12] M. A. O'Leary, D. A. Boas, B. Li, X. D. Chance, and A. G. Yodh, *Fluorescence lifetime imaging in turbid media*, Optics Letters **21**, 158 (1996).
- [13] S. Fantini, M. A. Franceschini, G. Gaida, E. Gratton, H. Jess, and W. W. Mantulin, *Frequency-domain optical mammography: edge effect corrections*, Med. Phys. **23**, 1 (1996).
- [14] J. van Houten, D. Benaron, S. Spilman, and D. Stevenson, *Imaging brain injury using time-resolved near-infrared light scanning*, Pediatric Research **39(3)**, 470 (1996).
- [15] S. Fantini, M. Franceschini, S. A. Walker, J. S. Maier, and E. Gratton, *Photon path distributions in turbid media: applications for imaging in Optical Tomography, Photon Migration and Spectroscopy of Tissue and Model Media; Theory, Human Studies and Instrumentation* (SPIE Optical Engineering Press, Bellingham, WA, 1995), Vol. 2389, pp. 340–349.
- [16] J. C. Schotland, J. C. Haselgrove, and J. S. Leigh, *Photon hitting density*, Applied Optics **32**, 448 (1993).
- [17] J. Chang, H. L. Graber, and R. L. Barbour, *Image reconstruction of dense scattering media from CW sources using constrained CGD and a matrix scaling*

- technique* in *Proceedings of Optical Tomography, Photon Migration, and Spectroscopy of Tissues and Model Media: Theory, Human Studies, and Instrumentation* (SPIE Optical Engineering Press, Bellingham, WA, 1995), pp. 682–691.
- [18] M. A. O’Leary, D. A. Boas, B. Chance, and A. G. Yodh, *Experimental images of heterogeneous turbid media by frequency-domain diffusing-photon tomography*, *Optics Letters* **20**, 426 (1995).
- [19] S. Arridge, P. van der Zee, M. Cope, and D. Delpy, *Reconstruction method for infrared absorption imaging in Time Resolved Spectroscopy and Imaging of Tissues* (SPIE Optical Engineering Press, Bellingham, WA, 1991), Vol. 1431, p. 204.
- [20] B. W. Pogue, M. S. Patterson, and T. J. Farrell, *Forward and inverse calculations for frequency domain diffuse optical tomography* in *Optical Tomography, Photon Migration and Spectroscopy of Tissue and Model Media; Theory, Human Studies and Instrumentation* (SPIE Optical Engineering Press, Bellingham, WA, 1995), Vol. 2389, p. 328.
- [21] C. L. Hutchinson, T. L. Troy, and E. M. Sevick-Muraca, *Fluorescence lifetime spectroscopy and imaging in random media* in *Optical Tomography, Photon Migration and Spectroscopy of Tissue and Model Media; Theory, Human Studies and Instrumentation* (SPIE Optical Engineering Press, Bellingham, WA, 1995), Vol. 2389, p. 274.
- [22] D. A. Boas, M. A. O’Leary, B. Chance, and A. G. Yodh, *Detection and characterization of optical inhomogeneities with diffuse photon density waves: a signal to noise analysis.*, *J. Opt. Soc. A*, special edition of Diffusing Light Probes of Turbid Media, accepted for publication (1996).
- [23] C. L. Matson, N. Clark, L. McMackin, and J. S. Fender, *Three-dimensional tumor localization in thick tissue using measurements in a plane* in *Advances in Optical Imaging and Photon Migration* (Opt. Soc. Am., Washington D.C., 1996).

- [24] J. S. Schotland, *Inversion formulae for diffusion imaging in Optical Tomography, Photon Migration and Spectroscopy of Tissue and Model Media; Theory, Human Studies and Instrumentation* (SPIE Optical Engineering Press, Bellingham, WA, 1995), Vol. 2389, pp. 304–310.
- [25] M. Ishii, J. S. Leigh, and J. Schotland, *Photon diffusion imaging in model and biological systems in Optical Tomography, Photon Migration and Spectroscopy of Tissue and Model Media; Theory, Human Studies and Instrumentation* (SPIE Optical Engineering Press, Bellingham, WA, 1995), Vol. 2389, p. 312.
- [26] D. Pattanayuk, General Electric Consumer Research and Development Report on *Resolution of optical images formed by diffusive waves in highly scattering media*.
- [27] A 3 mW, 780 nm source is modulated at 200 MHz. The turbid medium is 6.0 cm thick with a reduced scattering coefficient,  $\mu'_s$ , of  $10.0 \text{ cm}^{-1}$  and  $\mu_a = 0.05 \text{ cm}^{-1}$ . An spherical object is embedded in the middle of the slab. Light is collected and delivered to a photo-multiplier tube via an optical fiber with a diameter of 0.4 cm. For the simulations, the source and detector are scanned together along the boundary or the source is held fixed close to the object and the detector is scanned. Two different objects are studied; an absorbing object with  $\mu'_{s,in} = 10.0 \text{ cm}^{-1}$  and  $\mu_{a,in} = 0.15 \text{ cm}^{-1}$  and a scattering object with  $\mu'_{s,in} = 15.0 \text{ cm}^{-1}$  and  $\mu_{a,in} = 0.05 \text{ cm}^{-1}$ . For a positional uncertainty of  $10 \text{ }\mu\text{m}$  in either the source or the detector, the fractional error in the amplitude is  $2 \times 10^{-3}$  and the phase noise is  $0.03^\circ$ .
- [28] M. A. O'Leary, D. A. Boas, B. Chance, and A. G. Yodh, *Images of inhomogeneous media using diffuse photon density waves in Advances in Optical Imaging and Photon Migration* (Opt. Soc. Am., Washington D.C., 1994), Vol. 21, pp. 106–115.

- [29] D. A. Boas, L. E. Campbell, and A. G. Yodh, *Scattering and imaging with diffusing temporal field correlation*, Phys. Rev. Lett. **75**, 1855 (1995).
- [30] B. Beauvoit, S. Evans, T. Jenkins, E. Miller, and B. Chance, *Correlation between the light-scattering and the mitochondrial content of normal-tissues and transplantable rodent tumors*, Analytical Biochemistry **226(1)**, 167 (1995).
- [31] B. Beauvoit, T. Kitai, H. Liu, and B. Chance, *Time-resolved spectroscopy of mitochondria, cells, and rat tissues under normal and pathological conditions*, SPIE Optical Engineering Press **2326**, 127 (1993).
- [32] M. Kohl, M. Cope, M. Essenpreis, and D. Bocker, *Influence of glucose-concentration on light-scattering in tissue-simulating phantoms*, Optics Letters **19(24)**, 2170 (1994).
- [33] S. B. Bambot, J. R. Lakowitz, and G. Rao, *Potential applications of lifetime-based phase-modulation fluorimetry in bioprocess and clinical monitoring*, Trends in Biotechnology **13(3)**, 106 (1993).
- [34] J. B. Fishkin, O. Coquoz, E. R. Anderson, M. Brenner, and B. J. Tromberg, *Frequency-domain photon migration measurements of normal and malignant tissue optical properties in a human subject*, J. Opt. Soc. A, accepted (1996).
- [35] The Intralipid used here can be obtained from The Hospital of the University of Pennsylvania, Philadelphia, Pennsylvania.
- [36] H. van Staveren, C. J. M. Moes, J. van Marle, S. A. Prahl, and J. C. van Gemert, *Light scattering in intralipid-10% in the wavelength range of 400-1100 nm*, Applied Optics **30(31)**, (1991).
- [37] E. Hecht, *Optics* (Addison-Wesley, Reading, MA, 1987), Chap. 4, 5, 6.
- [38] J. Weng, M. Z. Zhang, K. Simons, and B. Chance, *Measurement of biological tissue metabolism using phase modulation spectroscopic technology in Time Re-*

- solved Spectroscopy and Imaging of Tissues* (SPIE Optical Engineering Press, Bellingham, WA, 1991), Vol. 1431, p. 161.
- [39] As 780 nm light fields with an optical frequency  $\omega \approx 10^{16} Hz$  impinge on a photocathode, the emission of photoelectrons is proportional to the square of the field. The square of the field has a part oscillating at  $2\omega$  and a dc part. In our case the detector speed is much slower than  $1/2\omega$ , so oscillations average to zero, and we only measure the dc part. For this reason we are able to ignore the optical frequency in our calculation of the measured intensity.
- [40] There are factors such as the quantum efficiency of the photocathode, the actual gain of the dynode chain, etc. which we do not attempt to measure. Instead we either calibrate the device on a known model system or make relative measurements. In addition, there are phase shifts induced as the modulated light passes through an optical fiber, and as electric signals pass through capacitors and lengths of cable.
- [41] Photodiodes, Hamamatsu Photonics K.K., Solid State Division, Japan (1995).
- [42] S. Yokoyama, A. Okamoto, A. Tsutomu, and S. Norihito, *Examination to eliminate undesirable phase delay of an avalanche photodiode (APD) for intensity modulated light*, Rev. Sci. Instr. **66(11)**, 5331 (1995).
- [43] The frequency drift of the phase locked Rhode & Schwartz generators drift is less than we are capable of measuring. The upper limit on the phase locked generators (measured with the SRS530) is 0.1 kHz/hour drift.
- [44] Mini-Circuits, *RF/IF Designer's Handbook*, New York, 1992/93.
- [45] A. G. Yodh and B. Chance, *Spectroscopy and imaging with diffusing light*, Physics Today **48(3)**, 34 (1995).
- [46] *Advances in Optical Imaging and Photon Migration* (Opt. Soc. Am., Washington D.C., 1994), Vol. 21.



- [47] *Medical Optical Tomography: Functional Imaging and Monitoring* (SPIE Optical Engineering Press, Bellingham, WA, 1993), Vol. Is11.
- [48] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging* (IEEE Press, New York, 1988).
- [49] This is not true at the source position. Thus we have assumed here that we will not be reconstructing the region containing the source.
- [50] M. S. Patterson and B. W. Pogue, *Effect of finite volumes on noninvasive frequency domain measurements of tissue properties* in *Advances in Optical Imaging and Photon Migration* (Opt. Soc. Am., Washington D.C., 1994), Vol. 21, p. 96.
- [51] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing* (Cambridge University Press, New York, 1988).
- [52] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations* (Prentice Hall, Englewood Cliffs, N.J., 1977), Chap. 9.
- [53] C. P. Gonatas, personal communication.
- [54] K. Tanabe, , *Numerical Mathematics* **17**, 203 (1971).
- [55] M. Firbank, M. Hiraoka, and D. Delpy, *A design for a stable and reproducible phantom for use in near infra-red imaging and spectroscopy*, *Phys. Med. Biol.* **38**, 847 (1993).
- [56] S. L. Jaques, M. R. Ostermeyer, L. Wang, and A. H. Hielscher, *Effects of sources, boundaries and heterogeneities on photon migration* in *Advances in Optical Imaging and Photon Migration* (Opt. Soc. Am., Washington D.C., 1994), Vol. 21, p. 83.

- [57] R. L. Barbour, H. L. Graber, J. W. Chang, S. L. S. Barbour, P. C. Koo, and R. Aronson, *MRI-guided optical tomography—prospects and computation for a new imaging method*, IEEE Computational Science & Engineering **2(4)**, 63 (1995).
- [58] M. Schweiger and S. R. Arridge, *Near-infrared imaging: photon measurement density functions in Optical Tomography, Photon Migration and Spectroscopy of Tissue and Model Media; Theory, Human Studies and Instrumentation* (SPIE Optical Engineering Press, Bellingham, WA, 1995), Vol. 2389, pp. 366–388.
- [59] S. Arridge, M. Cope, and D. T. Delpy, *The theoretical basis for the determination of optical pathlengths in tissue: a temporal and frequency domain analysis*, Phys. Med. Biol. **37(7)**, 1531 (1992).
- [60] R. C. Haskell, L. O. Svaasand, T. Tsay, T. Feng, M. S. McAdams, and B. J. Tromberg, *Boundary conditions for the diffusion equation in radiative transfer*, J. Opt. Soc. Amer. A **11**, 2727 (1994).
- [61] S. Glasstone and M. C. Edlund, in *The Elements of Nuclear Reactor Theory* (D. van Nostrand Co., New York, 1952), Chap. 5 & 14.
- [62] K. M. Case and P. F. Zweifel, in *Linear Transport Theory* (Addison-Wesley, MA, 1967), Chap. 8.
- [63] R. Aronson, *Boundary conditions for diffusion of light*, J. Opt. Soc. Amer. A **12**, 2532 (1995).
- [64] K. M. Case and P. F. Zweifel, *Linear Transport Theory* (Addison-Wesley, MA, 1967).
- [65] J. D. Jackson, *Classical Electrodynamics* (John Wiley & Sons, New York, NY, 1975).
- [66] J. Virmont and G. Ledanois, *Near-infrared medical imaging direct and inverse problem in Advances in Optical Imaging and Photon Migration, Technical Digest* (Opt. Soc. Am., Washington D.C., 1996).

- [67] E. M. Sevick-Muraca and C. L. Burch, *Origin of phosphorescence signals reemitted from tissues*, Optics Letters **19**(23), 1928 (1994).
- [68] M. S. Patterson and B. W. Pogue, *Mathematical-model for time-resolved and frequency-domain fluorescence spectroscopy in biological tissue*, Applied Optics **33**(10), 1963 (1994).
- [69] A. Knüttel, J. M. Schmitt, R. Barnes, and J. R. Knutson, *Acousto-optic scanning and interfering photon density waves for precise localization of an absorbing (or fluorescent) body in a turbid medium*, Rev. Sci. Instrum. **64**, 638 (1993).
- [70] X. D. Li, B. Beauvoit, R. White, S. Nioka, B. Chance, and A. G. Yodh, *Tumor localization using fluorescence of indocyanine green in Optical Tomography, Photon Migration and Spectroscopy of tissue and Model Media; Theory, Human Studies and Instrumentation* (SPIE Optical Engineering Press, Bellingham, WA, 1995), Vol. 2398, p. 789.
- [71] J. Wu, Y. Wang, L. Perelman, I. Itzkan, R. R. Dasari, and M. S. Feld, *Time-resolved multichannel imaging of fluorescent objects embedded in turbid media*, Optics Letters **20**, 489 (1995).
- [72] W. L. Rumsey, J. M. Vanderkooi, and D. F. Wilson, *Imaging of phosphorescence: a novel method for measuring oxygen distribution in perfused tissue*, Science **241**, 1649 (1988).
- [73] H. Szmecinski and J. R. Lakowicz, *Optical measurements of pH using fluorescence lifetimes and phase-modulation fluorometry*, Analytical Chemistry **65**, 1668 (1993).
- [74] J. R. Lakowicz, H. Szmecinski, K. Nowaczyk, and M. L. Johnson, *Fluorescence lifetime imaging of calcium using Quin-2*, Cell Calcium **13**, 131 (1992).
- [75] J. Folkman, *Introduction: Angiogenesis and cancer*, seminars in Cancer Biology **3**, 65 (1992).

- [76] P. W. Vaupel, S. Frinak, and H. I. Bicher, *Heterogeneous oxygen partial pressure and pH distribution in C3H mouse mammary adenocarcinoma*, *Cancer Research* **41**, 2008 (1981).
- [77] C. L. Hutchinson, J. R. Lakowicz, and E. Sevick-Muraca, *Fluorescence lifetime-based sensing in tissues: a computational study*, *Biophysical Journal* **68**, 1574 (1995).
- [78] J. Wu, M. S. Feld, and R. P. Rave, *Analytical model for extracting intrinsic fluorescence in turbid media*, *Applied Optics* **32(19)**, 3585 (1993).
- [79] X. D. Li, M. A. O'Leary, D. A. Boas, B. Chance, and A. G. Yodh, *Fluorescent diffuse photon-density waves in homogeneous and heterogeneous turbid media - analytic solutions and applications*, *Applied Optics* **35(19)**, 3746 (1996).
- [80] Cardio-Green is a water soluble, tricarboyanine dye. We obtained the dye from Hynson, Wescott Dunning, Inc., Baltimore, Md. 21201, USA.
- [81] S. Zhau, M. A. O'Leary, S. Nioka, and B. Chance, *Breast tumor detection using continuous wave light sources in Optical Tomography, Photon Migration and Spectroscopy of tissue and Model Media; Theory, Human Studies and Instrumentation* (SPIE Optical Engineering Press, Bellingham, WA, 1995), Vol. 2398, pp. 809–817.
- [82] Clinical breast imaging data are expected to be published soon by S. B. Colak, Philips Research Labs, Netherlands.
- [83] B. J. Tromberg, S. Madsen, C. Chapman, L. O. Svaasand, and R. C. Haskell, *Frequency-Dominant photon migration in turbid media in Advances in Optical Imaging and Photon Migration* (Opt. Soc. Am., Washington D.C., 1994), Vol. 21.