University of Cape Town

Masters Dissertation

4D Flow and Displacement Sensitive MR Imaging of Upper-Arm Arteriovenous Connections for Haemodialysis

Author:
Stephen Jermy

Supervisor:
Prof. Ernesta Meintjes
Assoc. Prof. Thomas Franz
Dr. Daniel Auger

A thesis submitted in fulfilment of the requirements
for the degree of Master of Science (Medicine) in Biomedical Engineering

in the

Division of Biomedical Engineering
Department of Human Biology

February 2016

Declaration of Authorship

I, Stephen Jermy, hereby declare that the work on which this dissertation is based is my original work (except where acknowledgements indicate otherwise) and that neither the whole work nor any part of it has been, is being, or is to be submitted for another degree in this or any other university.

I empower the university to reproduce for the purpose of research either the whole any portion of the contents in any manner whatsoever.

Signed: _____

Date: _____

Acknowledgements

I would like to sincerely thank my supervisors who helped me immeasurably during the course of my Masters project. I huge thank you to Prof. Ernesta Meintjes for her encouragement and insight that helped me see my way forward when a myriad of struggles arose; to Assoc. Prof. Tom Franz for his help and guidance during the clinical aspects of the project; and to Dr. Daniel Auger for his help during the early stages of my project. Additionally, I would like to thank Bruce Spottiswoode for all his help and encouragement during his short visits; and to my colleagues and friends with whom I had the pleasure of sharing Ernesta's lab space for all the fantastic conversations. Finally, I would like to thank my family for the all things they have done for me, I am endlessly grateful.

This dissertation is based upon work supported financially by the National Research Foundation of South Africa.

UNIVERSITY OF CAPE TOWN

Abstract

Faculty of Health Sciences
Department of Human Biology

Master of Science (Medicine) in Biomedical Engineering

4D Flow and Displacement Sensitive MR Imaging of Upper-Arm Arteriovenous Connections for Haemodialysis

by Stephen JERMY

Chronic Kidney Disease (CKD) is a disease that causes kidney damage, often leading to the patient requiring haemodialysis treatment. Haemodialysis treatment requires a vascular access method, commonly Arteriovenous (AV) fistulae and grafts. These access methods must be regularly assessed to ensure the access remains unblocked and the flow rate is normal. Phase Contrast MRA (PC-MRA) is a versatile Magnetic Resonance Imaging (MRI) modality which is capable of imaging and quantifying blood flow in vivo. It is for this reason that this imaging technique was used to image blood flow in the vasculature of the upper arm of volunteers and haemodialysis patients with either an AV fistula or graft. This imaging technique is capable of producing temporally resolved Three-dimensional (3D) datasets (known as "Four-dimensional (4D)" flow) of blood flow in major vessels. Velocities are phase encoded between $-\pi$ and π based on the chosen Velocity Encoding Constant (v_{enc}). To successfully characterise all velocities in the volume it is necessary to set the v_{enc} to be approximately equal to the highest velocity found in the vessel. Any lower v_{enc} value will cause phase wrapping, an imaging artefact causing all higher velocities to be wrapped by a multiple of 2π. However, the increase in sensitivity to high velocities reduces the overall specificity of the velocities, especially for low velocities. Due to the pulsatile nature of blood flow in arterial vessels, a large range of velocities are encountered, while venous flow is more constant but lower than the peak arterial flow value. For this reason and due to the length of the 4D flow scans, 20-30 minutes, it would be preferable to perform one scan at a relatively low v_{enc} and correct any phase wrapping during post-processing. In this study, we performed both Two-dimensional (2D) PC-MRA scans at various locations in the upper arm and 4D PC-MRA scaans with similar v_{enc} settings. The purpose of the study was to implement and test several methods of phase unwrapping to remove phase wrapping artefacts from affected areas within the PC-MRA datasets.

Using a 1.5T MRI scanner (MAGNETOM Symphony, Siemens AG, Erlangen), blood flow from vessels of the upper arm were collected from five healthy volunteers (Age: 27.2 ± 3.82 years) and eight haemodialysis patients (Age: 36.5 ± 11.6 years), of which five patients had brachiocephalic fistulae (Age: 35 ± 14.7 years) and three patients had brachiocephalic grafts (Age: 39 ± 9.17 years). In total, nine volunteer and 14 patient 2D PC-MRA datasets, and eight volunteer and five patient 4D PC-MRA datasets were selected to be unwrapped. All protocols were approved by the Human Research Ethics Committee. The images were processed to remove noise and eddy currents.

Several methods of phase unwrapping were investigated and implemented in order to remove phase wrapping artefacts from the 2D and 4D PC-MRA datasets. The methods investigated included temporal phase unwrapping, the Goldstein-Zebker-Werner branch cut method, Nearest Neighbour Algorithms (NNAs), Quality Guided Phase Unwrapping (QGPU), and a 3D branch cut method. The temporal phase unwrapping algorithm is performed along one axis only, the temporal axis, and as such is highly sensitive to noise. The Goldstein branch cut method, NNA, and QGPU are all 2D spatial phase unwrapping algorithms that were able to overcome the problems associated with image noise by detecting and avoiding phase residues. The 3D branch cut method extends these 2D methods, especially the NNA, to create branch cut surfaces that were used to avoid Phase Singularity Loops (PSLs). All of the phase unwrapping algorithms require a starting pixel value that is already unwrapped to use as a reference. If a pixel is chosen that does not fulfil this requirement all subsequent "unwrapped" pixels will be offset by some integer multiple of 2π. All software was developed and implemented using MATLAB (Mathworks, Natick, MA) on an Intel Core i7 with 16 GB of RAM.

The temporal phase unwrapping, Goldstein branch cut, and Phase Derivative Variance (PDV) QGPU algorithms were used to unwrap the 2D PC-MRA datasets that contained phase wrapping artefacts. A new spatio-temporal algorithm that combined facets of temporal phase unwrapping and the Goldstein branch cut method was developed and was also used to unwrap these datasets. All methods were first validated to determine their effectiveness with various Signal to Noise Ratios (SNRs), v_{enc} settings, Region of Interest (ROI) widths, and Repetition Times (TRs). The temporal phase unwrapping algorithm, due to its unique limitations, could only be applied to the volunteer datasets and was able to unwrap 2 of 9 volunteer datasets (22.2%) successfully. The Goldstein branch cut and QGPU algorithms unwrapped 3 of 9 volunteer (33.3%) and 5 of 14 patient datasets (35.7%), and 3 of 9 volunteer (33.3%) and 4 of 14 patient datasets (28.6%), respectively. The new spatio-temporal algorithm was able to unwrap 3 of 9 volunteer (33.3%) and 8 of 14 patient datasets(57.1%). The 3D branch cut method was also used to unwrap the 4D PC-MRA datasets that contained phase wrapping artefacts. The 3D branch cut method was able to successfully unwrap 5 of 8 volunteer (62.5%) and 4 of 5 patient datasets (80.0%).

The phase unwrapping algorithms were implemented correctly and applied to the PC-MRA data. The newly developed spatio-temporal algorithm performed the best of the 2D phase unwrapping methods however this is partially due to the unique nature of the patient flow data. The 3D branch cut method was able to perform well even on datasets with relatively low SNRs. The most common reason that a phase unwrapping algorithm would fail to successfully unwrap a PC-MRA dataset was because of low spatial resolution and low v_{enc} settings.

Contents

List of Figures

List of Tables

Acronyms

e-PTFE	expanded Polytetrafluoroethylene
pmp	per million population
v_{enc}	Velocity Encoding Constant
AV	Arteriovenous
CKD	Chronic Kidney Disease
DDU	Duplex Doppler Ultrasound
DSA	Digital Subtraction Angiography
ESRD	End Stage Renal Disease
FE	Flow Encoding
FID	Free Induction Decay
GFR	Glomerular Filtration Rate
GRE	Gradient Echo
KDOQI	Kidney Disease Outcomes Quality Initiative
MNNA	Modified Nearest Neighbour Algorithm
MRA	Magnetic Resonance Angiography
MRI	Magnetic Resonance Imaging
NKF	National Kidney Foundation
NMR	Nuclear Magnetic Resonance

NNA Nearest Neighbour Algorithm

QGPU Quality Guided Phase Unwrapping

PC-MRA Phase Contrast MRA

PDV Phase Derivative Variance

PSL Phase Singularity Loop

RF Radiofrequency

ROI Region of Interest

RRT Renal Replacement Therapy

SNR Signal to Noise Ratio

TE Echo Time

TR Repetition Time

WPG Weighted Phase Gradient

1D One-dimensional

2D Two-dimensional

3D Three-dimensional

4D Four-dimensional

Physical Constants

Proton-gyromagnetic ratio	γ_p	=	$2.675\ 222\ 005(63) \times 10^8$	$\mathrm{s}^{-1}\ \mathrm{T}^{-1}$
	$\gamma\!\!\!^{-}_p$	=	$42.577\ 480\ 6(10)$	$\mathrm{MHz}\ \mathrm{T}^{-1}$

Symbols

B_0	Large external magnetic field	T
B_1	Magnetic component of RF pulse	T
M_0	Net magnetisation vector	A/m
$M_{xy}(t)$	Transverse magnetisation	A/m
$M_z(t)$	Longitudinal magnetisation	A/m
v_{enc}	Velocity encoding parameter	cm/s
α	Flip angle	rad
ω_0 or f_0	Larmor frequency	rad/s or Hz

Chapter 1

Introduction

1.1 Background to Study

Chronic Kidney Disease (CKD) is a disease that adversely affects kidney function. The disease is characterised by a sustained loss of renal function over the course of several months and may lead to End Stage Renal Disease (ESRD), a life threatening disease characterised by almost complete renal failure. It is a serious risk factor for the development of cardiovascular disease, the leading cause of death in most of the developed world and steadily increasing in the developing world. A commonly cited statistic is that the estimated worldwide prevalence of CKD is at 10% of the global adult population (Meyers 2015, South African Renal Society 2006). Although there are no reliable statistics for the actual incidence and prevalence of CKD and ESRD, this estimate means that approximately 5 million South Africans are living with some stage of CKD or at least are at risk of developing CKD. In South Africa in the years 2007, 2010 and 2013 there were 6 523, 6 804 and 6 885 deaths with renal failure reported as the underlying cause of death, respectively, making it the 17th most common cause of death accounting for 1.7% of all deaths (Statistics South Africa 2013, 2014). Compared with the United States of America where the prevalence of CKD is at 6.3% of the population, approximately 17 million people (U.S. Renal Data System 2013*a*). In 2010 there were 44 362 deaths from renal failure, accounting for approximately 1.8% of all deaths (Murphy et al. 2013). Based on this it is reasonable to assume that the burden of disease related to renal failure in the USA is similar to that of South Africa. However, the largest discrepancy between the USA and South Africa is with the number of patients receiving treatment. In 1994 there

were 3 399 patients, 99 per million population (pmp), receiving treatment for ESRD in South Africa (Naicker 2003) which has since increased to 8 559 patients, 160 pmp, in 2012 (Meyers 2015). In the USA, however, there were approximately 250 000 patients, 950 pmp, receiving treatment for ESRD in 1994, and over 600 000 patients, 2 000 pmp, in 2010 (U.S. Renal Data System 2013*b*).

1.2 Purpose of Study

The most common treatment options for CKD are treatment of the underlying symptoms, Renal Replacement Therapy (RRT) and renal transplantation. The cost per patient for renal transplantation is R300 000 for the first year and R160 000-R180 000 for subsequent years while the cost per patient for RRT is R200 000 per year (Meyers 2015). These costs are often prohibitively expensive for hospitals and clinics operating in resource scarce environments leading to South Africa's lack of treatment availability. The following are the most common treatment options for patients with CKD.

- **Medication for the treatment of underlying symptoms**
  - Requires relatively high kidney function, i.e. early stage CKD patients.

  (Clement et al. 2009, U.S. Renal Data System 2013*a*)
- **Transplantation of donor kidney**
  - Generally reserved for ESRD patients
  - 2 year waiting list
  - Immunosuppressants required by patients to prevent host rejection

  (U.S. Renal Data System 2013*a*,*b*)
- **Renal Replacement Therapy**

  Peritoneal dialysis
  - Less popular RRT

  (U.S. Renal Data System 2013*a*)

  Haemodialysis
  - See Section 2.1.1

- Requires vascular access

Patients that do receive RRT for CKD, especially haemodialysis patients, will require surgery to create a vascular access for the haemodialysis machine. The most common vascular access methods are the creation of Arteriovenous (AV) fistulae and grafts. These access methods are highly susceptible to occlusion, especially because accelerated atherosclerosis is a common complication associated with CKD (Shoji et al. 2012, Tsimihodimos et al. 2011), and are prone to failure in these circumstances (Ascher & Hingorani 2004, Gilpin & Nichols 2010, Hung et al. 2010, Mansilla et al. 2001, Matsuura et al. 1998). Further noted by Ascher & Hingorani (2004) is the recommendation that preemptive tests be conducted to determine the flow through the fistulae and grafts to help improve the access method's patency.

Magnetic Resonance Imaging (MRI) is well suited to the imaging of blood flow in vessels (Bushberg et al. 2001, Kim & Parker 2012) as well as the ability to actually measure the flow values (Buonocore 1998, Markl et al. 2012, 2011, Wigström et al. 1996). This information can be used to determine flow characteristics in the blood vessels.
For this reason MRI was used as the primary imaging modality for this project with the aims of implementing methods for imaging blood flow in haemodialysis access vessels as well as to develop tools to analyse data from MRI flow scans and extract important blood flow characteristics.

1.2.1 Scope and Limitations

The MRI data was gathered using a 1.5T Full-body MRI scanner (MAGNETOM Symphony, Siemens AG, Erlangen). 5 healthy volunteers (Age: 27.6±3.82 years) and 8 patients (Age: 36.5±11.6 years) were used as subjects during the course of this project. Of the 8 patients undergoing haemodialysis treatment, 5 patients had AV fistulae (Age: 35±14.7 years) and 3 patients had AV grafts (Age:39±9.17 years). The data acquisition took place over the course of two years, from 2013 to 2014.

1.3 Overview of Study

Chapter 2 gives an overview of the anatomy of the kidney and a description of Chronic Kidney Disease (CKD). Common treatment options are discussed with specific interest paid to haemodialysis and its associated access methods. Common modalities of vascular imaging are discussed, leading to the use of MRI in the imaging of blood flow. The specific techniques for imaging blood flow used in MRI are described as well as issues related to these techniques, specifically phase wrapping.

Chapters 3 and 4 discuss the methodology used for acquiring and preprocessing Phase Contrast MRA (PC-MRA) images and the techniques used to remove phase wrapping artefacts from these images, respectively. The materials and techniques used to acquire and process the PC-MRA images are discussed and the results of the acquisitions are given. Several methods of performing phase unwrapping are investigated and implemented as well as a method for performing a validation of these methods.

Chapter 5 reports the results of the various unwrapping techniques used. The techniques include temporal phase unwrapping, the Goldstein-Zebker-Werner branch cut method, Phase Derivative Variance (PDV) Quality Guided Phase Unwrapping (QGPU), and a 3D branch cut method. The results of both the validation and the actual testing are given.

Chapter 6 discusses the results from the previous chapter and determine how effective each phase unwrapping technique was.

Finally, Chapter 7 provides the conclusions made based on the previous chapters. Recommendations for possible future work are also detailed.

Chapter 2

Literature Review

2.1 Chronic Kidney Disease

The kidneys are a pair of organs found in the abdominal cavity and are used in the human body to remove various waste products found in blood due to the body's metabolic processes. Figure 2.1 shows the location of the kidneys in the human body. The kidneys are found against the posterior muscular wall to the left and right of the vertebral column. They are bounded superiorly by the diaphragm and anterosuperiorly by the liver and the spleen, for the right and the left kidney respectively. The kidneys are surrounded by layers of fat that act as protection for the kidneys.

The internal structure of the kidney is shown in Figure 2.2a and is composed of two distinct regions known as the outer cortex and the inner medulla. The cortex surrounds the medulla and separates the renal pyramids with projections known as renal columns; the cortex contains a large bed of capillaries. The medulla is split into between 8 and 15 pyramidal structures, called renal pyramids, that are striped in appearance as they are composed of small tubules and blood vessels. A renal pyramid is shown in Figure 2.2b. The tip of the renal pyramid projects into a small space, a minor calyx, which in turn is grouped with nearby minor calyces to form a major calyx. The major calyces form an area known as the renal pelvis where urine is collected and transported away from the kidneys.

Within the cortex and extending into the medulla are millions of tubular structures, each of these structural units is known as a nephron, shown below in Figure 2.2c. The

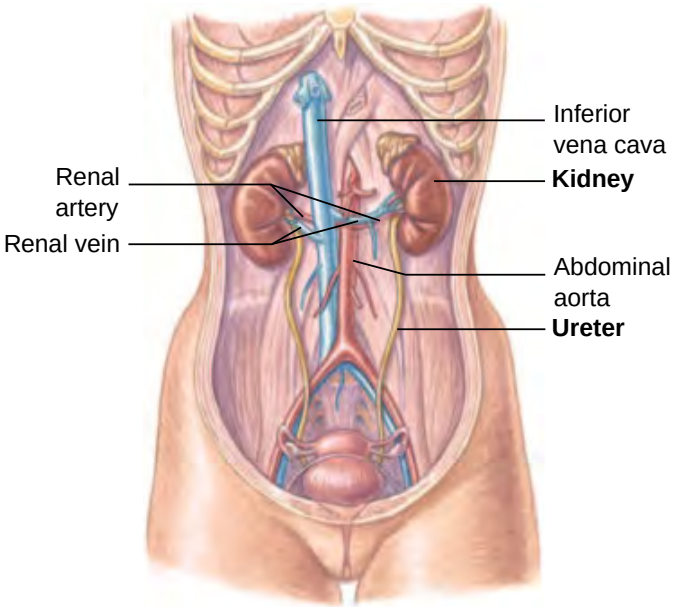
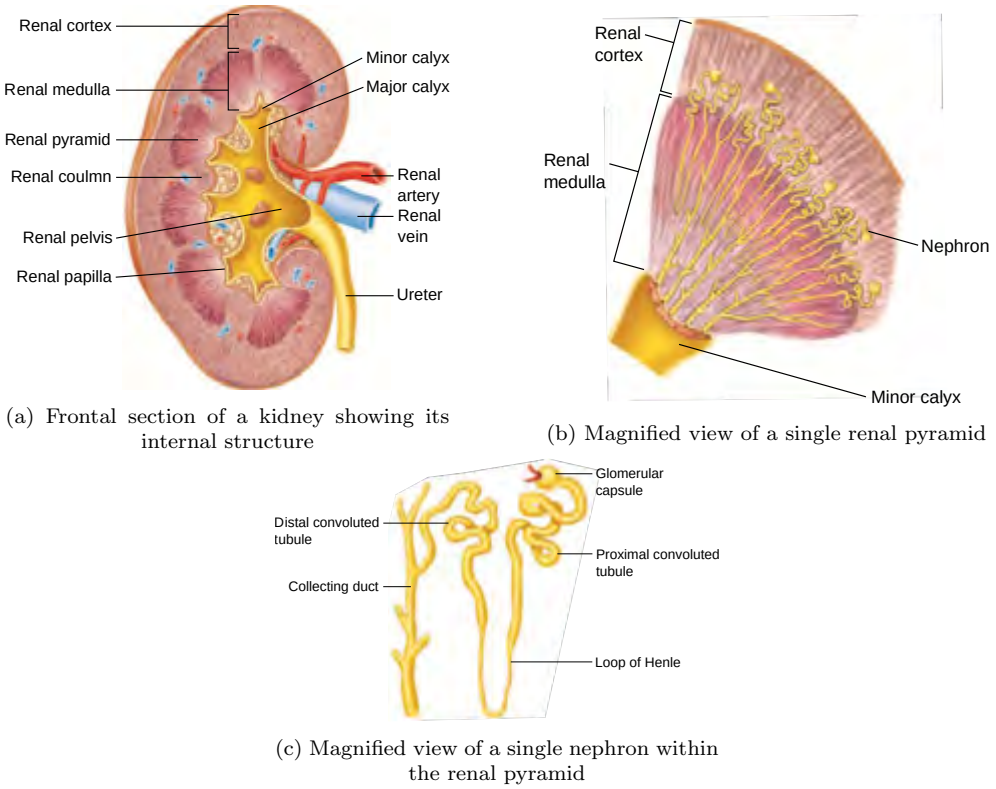


Figure 2.1: Anterior view of the kidneys in situ, the digestive tract has been removed for clarity. Image adapted from Fox 2011

cortex contains the head of the nephron, the renal corpuscle, which surrounds a capillary network. Solutes and water found in the blood plasma are filtered into the corpuscle which enters a network of tubules. The filtrate is modified into urine which is collected in a collecting duct and transported out of the nephron. During this process vital nutrients, such as sodium ions, glucose, and water are reabsorbed and returned to the bloodstream.

Chronic Kidney Disease (CKD) is a term that describes prolonged and continuous loss of kidney function. CKD is divided into five stages which are identified by damage to the kidneys and the patient's Glomerular Filtration Rate (GFR), an estimate of the flow rate of blood through the capillaries in the renal corpuscle. GFR is measured in units of ml/min/1.73m^2, the 1.73m^2 factor is the average surface area of a human body, and the accepted normal level is greater than 90 ml/min/1.73m^2. Stage 1 and 2 are the least severe, with kidney damage and little to no reduction in GFR, while stage 5 indicates renal failure, commonly known as End Stage Renal Disease (ESRD) (Shoji et al. 2012) Early stages of CKD are often asymptomatic and are usually detected by urinalysis, specifically an increase of protein concentration in urine. As the disease progresses symptoms begin to develop such as hypertension, due to retention of salt; acidosis and hyperkalemia, due to elevated concentrations of ionic hydrogen and potassium, respectively; and accelerated atherosclerosis which can lead to cardiovascular disease (Lowth 2013). There are several known causes and risk factors for CKD; the three



(a) Frontal section of a kidney showing its internal structure

(b) Magnified view of a single renal pyramid

(c) Magnified view of a single nephron within the renal pyramid

Figure 2.2: The internal structure of the kidney. Images adapted from Fox 2011

most common causes are diabetes mellitus, hypertension, and glomerulonephritis which accounts for approximately 75% of all cases (Fox 2011, chap. 17)(Lowth 2013).

There are several treatment options for CKD, depending on the stage, which largely focus on slowing the loss of renal function, reducing the risk of cardiovascular disease, and treating any complications (Drawz et al. 2009). The most common treatment options for late stage CKD are Renal Replacement Therapy (RRT) and transplantation (Lowth 2013).

2.1.1 Haemodialysis

RRT is used in more than 60% of all cases of CKD as the primary treatment option (Gibbons 2006). In fact patients with ESRD will always require some level of medical intervention. The purpose of RRT is to effectively replace the kidneys with external equipment that is able to replicate the function of filtering blood and removing waste products from the body (Ostermann 2011).

Haemodialysis is the most common method of RRT and is indicated for a patient with a GFR of less than 15 ml/min/1.73m^2. Unfiltered blood is continuously extracted from an

artery in the arm into a dialysis machine, the machine uses diffusion to remove waste as well as excess water and salt. The cleansed blood is then recirculated into the patient's body through a vein in the arm. Patients can either undergo haemodialysis continuously or intermittently as evidence suggests that both methods are effective (Ostermann 2011). Continuous haemodialysis is usually reserved for bedridden patients due to its continuous nature whereas intermittent haemodialysis is usually performed three times a week for between four and six hours. The largest difference between the two modes is the required blood flow for the dialysis machine to function optimally, 100-250 and 200-350 ml/min for continuous and intermittent haemodialysis, respectively (Ostermann 2011).

The mean blood flow rates for the brachial artery and cephalic vein are approximately 170 (Ozcan et al. 2006) and 28 ml/min (Albayrak et al. 2007), respectively. To allow these high blood flow rates in these vessels it is necessary to introduce some form of access that can withstand high blood flow. Initially central venous catheters are inserted into the patient's blood vessels but these are removed when a permanent access method is created, such as arteriovenous fistulas and grafts (Patel et al. 2011).

Arteriovenous Access

Arteriovenous (AV) fistulas and grafts are the two most common methods of haemodialysis access. AV fistulas are considered to be the superior method of access due to their comparatively lower risk of primary failure, increased primary patency, and lower rates of infection (Gibson et al. 2001, Jindal et al. 2006). The use of fistulas as the primary vascular access method has risen in the USA in the past decade where the most common method was the AV graft. The prevalence of fistulas as an access method for patient was 55% in 2007 while the prevalence of grafts was 27.2% in the same year (U.S. Renal Data System 2013*b*).

AV fistulas are surgically created by joining a vein and artery together, forming an anastomosis, while AV grafts are formed by connecting a vein and artery with either an autologous or synthetic graft. Several factors influence the placement of the fistula or graft. The diameter of the chosen vessels should be a minimum of 2 mm and 2.5 mm for arteries and veins, respectively. The anastomosis is placed as distally in the patient's non-dominant arm as possible, commonly, by connecting the radial artery and cephalic vein. Upper arm access is used as sites in the forearm either become unavailable or are found to be unsuitable. Common anastomosis configurations in the upper arm are

created between the brachial artery and either the cephalic vein or basilic vein, although the basilic vein needs to be at least 4 mm in diameter. Generally after no suitable sites are found in the non-dominant arm, an AV graft can be used in the same configurations as the AV fistula. Finally if no suitable sites can be found, the dominant arm as well as the lower limbs can be investigated for sites (Gilpin & Nichols 2010).

AV fistulas and grafts can only be used for haemodialysis after a certain period of time, postoperatively, during which the access site undergoes maturation. Fistulas mature over a period of two to three months during which the walls of the veins expand and thicken greatly to support the increased blood flow, usually between 300 and 600 ml/min, this improves the patency as low flow rates in the anastomosis are correlated with clot formation which can lead to stenotic formations, further reducing the blood flow (Gilpin & Nichols 2010, Jindal et al. 2006). AV grafts require only three weeks before they can be used and during this period tissue proliferation occurs in the graft and maximum flow is achieved a few weeks after (Akoh 2009). The material most commonly used to make AV grafts, expanded Polytetrafluoroethylene (e-PTFE) (Patel et al. 2011), does not need to dilate and so can be used more rapidly than fistulas. However, this limits the blood flow to the diameter of the graft at its distal anastomosis (Akoh 2009). Intimal hyperplasia is common at the venous anastomosis of the graft which leads to stenosis and often restricted blood flow; should blood flow in the graft become lower than 600 ml/min a thrombus may begin to form in the graft further limiting blood flow (Akoh 2009).

The patency of a fistula and graft is its quality of remaining unblocked and is another important factor in choosing an appropriate access method for the patient. The patency of the access method is generally measured in terms of months and years and is divided into two categories, namely primary and secondary patency. Primary patency is measured as the amount of time between the creation of the access and the first time surgical intervention is necessary to maintain patency. Secondary patency is measured between the placement of the access and its permanent failure, that is its removal and replacement (Al-Jaishi et al. 2014). In Table 2.1 the primary and secondary patencies of AV fistulas and grafts have been collected from multiple sources. It shows that there is little definitive proof that one access method is superior to the other in terms of patency (Gibson et al. 2001). In fact the median survival for primary patency for the upper-arm AV fistulas and grafts is 3.36 and 4.38 years, respectively, and an additional 1.49 and 1.62 years, respectively, for second patency (Ates et al. 2006).

Table 2.1: Primary and secondary patencies of popular access methods

AV Fistula				AV Graft				
Primary Patency (%)		Secondary Patency (%)		Primary Patency (%)		Secondary Patency (%)		
1 year	2 years	1 year	2 years	1 year	2 years	1 year	2 years	**Ref**
-	-	-	-	-	-	59-90	47-85	Akoh (2009)
60	51	71	64	-	-	-	-	Al-Jaishi et al. (2014)
97.3 (6 mo.)	87.7	91 (6 mo.)*	71.4*	94.7 (6 mo.)	82.6	92.3 (6 mo.)*	77.6*	Ates et al. (2006)
51	38	-	-	-	-	-	-	Field et al. (2008)
70-85**	-	-	-	-	-	-	-	Gibbons (2006)
56.1	39.8	73.2	64.2	38.2	24.6	71.8	59.5	Gibson et al. (2001)
93	-	-	-	70	-	-	-	Gilpin & Nichols (2010)
49	39 (18 mo.)	70	67 (18 mo.)	-	-	-	-	Huijbregts et al. (2008)
-	-	-	-	31.9	17.4	-	-	Hung et al. (2010)
-	-	-	-	29	-	61.4	-	Mansilla et al. (2001)
-	-	-	-	68	46	80	51	Matsuura et al. (1998)

* Ates et al. (2006) defines secondary patency as the time between the first revision of the graft or fistula, to maintain patency, and its permanent failure
** Gibbons (2006) does not state whether this refers to primary or secondary patency

The National Kidney Foundation (NKF) first published a set of guidelines called the Kidney Disease Outcomes Quality Initiative (KDOQI) in 1997 which details several recommendations and best practices for treating and providing care for patients with CKD. These guidelines have been constantly updated and amended to reflect new findings from relevant literature. Their guidelines also make recommendations about vascular access for patients including preferred methods of access and access assessment. The KDOQI recommends that AV fistulas should be the access method of choice, followed by AV grafts, and finally (although it is stated that it should be avoided) a long-term catheter. The recommendation of the NKF is based on the slightly improved patency found in AV fistulas as well as the reduced complication rates (KDOQI 2006).

The KDOQI goes on to describe the guidelines for effective access assessment and surveillance. Recommendations are made to survey the access for signs of stenotic formations likely to cause disruption to blood flow; this coupled with surgical intervention can increase patency while reducing the chance of thrombosis (KDOQI 2006).

2.1.2 Vascular Imaging

The ability to directly, or if necessary indirectly, survey characteristics of vessels and blood flow within these vessels is important for both pre-operative selection of candidate vessels (Planken et al. 2007) as well as post-operative patient care (KDOQI 2006). There are several methods of vascular imaging that have been developed and used to assess blood vessels, for example, x-ray angiography, computed tomography angiography, Duplex Doppler Ultrasound (DDU), Digital Subtraction Angiography (DSA), and Magnetic Resonance Angiography (MRA) (Herrick & Hutchinson 2004, Planken et al.

2007). Not all of these methods are suited to imaging blood flow in haemodialysis accesses and as such the three most common, and supported, are DDU, DSA, and MRA.

DSA uses x-ray with and without iodinated contrast material to create a raw image and a subtraction mask, respectively, which together produce an image with high contrast between vessels and surrounding tissues. It is considered the gold standard for the assessment of vascular access (Planken et al. 2007). In general DSA is not able to quantify blood flow however Huang et al. (2013) was able to quantify blood flow from DSA images using an additional technique called the optical flow method.
DDU uses ultrasonography and the Doppler effect to produce coloured images of blood flow and can also give specific flow values. DDU has been recommended in KDOQI (2006) as the preferred method of vascular access assessment. There are, however, some drawbacks to using Doppler ultrasound due to cut-off limits on DDU units which can make it difficult or impossible to get accurate values (Planken et al. 2007).
MRA is a group of methods used to produce high contrast between vessels and blood, and surrounding areas using Magnetic Resonance Imaging (MRI). Several techniques have been developed that use contrast enhancement, similar to DSA, and non-contrast enhanced that make use of the physical properties of blood to produce contrast, for instance, time-of-flight (Bernstein et al. 2004, chap. 15) and Phase Contrast MRA (PC-MRA) (Planken et al. 2007).

2.2 Magnetic Resonance Imaging

MRI is a versatile imaging modality based on the spectroscopic discipline of Nuclear Magnetic Resonance (NMR). Its versatility is due to its sensitivity to a wide range of tissue properties as well as its relative safety compared with similar imaging modalities. NMR uses large magnetic fields and resonant Radiofrequency (RF) pulses to selectively excite the nuclei of specific atoms in some tissue. The properties of the tissue dictate how quickly the tissue can absorb and release the energy from the RF pulse which is used to create contrast between different tissues. MRI takes advantage of this to produce clinically relevant images of internal anatomical structures by spatially encoding and detecting the NMR signals. This allows the signal to be localised within the subject's body (Bushberg et al. 2001, chap. 14).

Due to the relatively high abundance of Hydrogen (^{1}H) in the human body, commonly as water, MRI is most often used to detect the magnetisation of its nucleus, a proton, hereafter referred to as its spin. ^{1}H has a large magnetic moment, however, a single spin is not large enough to be detected on its own. Furthermore, due to random thermal effects in tissue there is no net magnetisation under normal circumstances. To create a net magnetisation it is therefore necessary to apply a large, ideally homogeneous, magnetic field to the tissue. This large magnetic field, known as B_0, causes the spins to align in either a low or high energy state with a very slight majority of spins aligning in the low energy state. Therefore there is a net magnetisation, M_0, in the direction of the applied field (Bushberg et al. 2001, chap. 14).

Excitation The spins precess around the axis of B_0 at frequency f_0, known as the Larmor frequency. In Equation (2.1) it can be seen that the frequency is dependent on both the magnetic field strength and the unique gyromagnetic ratio associated with the imaged nucleus, γ. For ^{1}H, this is known as the proton-gyromagnetic ratio, γ_p. It is convenient to use the value γ_p, equal to $\gamma_p/2\pi$, as its units are MHz/T.

$$f_0 = \gamma B_0 \tag{2.1}$$

The spins do not precess in phase and so there is no net magnetisation in the transverse plane, the plane perpendicular to the applied field. A RF pulse, B_1, that is tuned to the specific Larmor frequency and applied perpendicularly to B_0, is used to apply energy to the spins in the low energy state to 'tip' them towards the high energy state. M_0 is tipped by a specific angle, known as the flip angle, α, dependent on the length and amplitude of the applied pulse. After the RF pulse has been applied the longitudinal component of magnetisation is given by $M_z = M_0 \cdot \cos\alpha$ while the transverse component of magnetisation is given by $M_{xy} = M_0 \cdot \sin\alpha$. This means that maximal transverse magnetisation is achieved at $\frac{\pi}{2}$ (90°). A coil tuned to the f_0 is then used to detect the M_{xy} signal which precesses around B_0 as a decaying sinusoid, known as the Free Induction Decay (FID) signal (Bushberg et al. 2001, chap. 14).

Recovery The reason for the decay of the FID signal is due to a loss of phase coherence caused by random fluctuations in the local magnetic field, known as spin-spin interaction

and is describe by the timing constant T_2. Inhomogeneities in the magnetic field can also cause a loss of phase coherence and are described by $T_2^{'}$. Together these factors are known as T_2^* and describe the rate at which M_{xy} exponentially decays. Another important timing constant, T_1, describes the exponential regrowth of the longitudinal signal as spins release energy into the surrounds and is known as spin-lattice relaxation (Brown et al. 2014, chap. 1). In general $2T_1 \geq T_2 \geq T_2^*$, however, when dealing with anatomical tissues it is accepted that $T_1 > T_2 > T_2^*$ (Levitt 2013). The tissues in the body have different timing constants which are determined by the size and movement of the molecules in the tissue. These timing constants are used to create contrast by controlling the Repetition Time (TR) and Echo Time (TE) of the pulse sequence. TR is the amount of time between consecutive excitation pulses, and TE is the time between the excitation pulse and the measurement of the signal (Bushberg et al. 2001, chap. 14). Blood flow can also be used to create contrast and is discussed in Section 2.2.1.

Signal localisation A fundamental aspect of creating the MR image is the ability to determine the location of the excited spins. To achieve spatial localisation it is necessary to introduce gradients to the B_0 field. Three sets of gradient coils are arranged so that they are orthogonal to one another, most frequently along the x-, y-, and z-axes. These can be switched on with different strengths in any combination at different times during the imaging sequence to achieve gradients in any direction. The small magnetic gradients set up non-uniformities in the homogeneous B_0 field which causes the spins to precess at different Larmor frequencies. A gradient applied perpendicularly to the imaging slice is known as the slice encode gradient and is used to selectively excite spins in a narrow portion of the patient's anatomy, called a slice. A gradient is then applied for a short time in either of two orthogonal in-plane directions, called the phase encode gradient, which introduces a phase shift in the spins based on their position. Finally, in the remaining in-plane axis, the frequency encode gradient alters the precessional frequencies of the spins within the slice while they are being read out. Using Fourier analysis methods the detected signal can be decomposed into its constituent parts and used to spatially register spins within the body (Bushberg et al. 2001, chap. 15).

2.2.1 Phase Contrast MRA

The branch of MRI that forms the focus of this project is Magnetic Resonance Angiography. MRA and its associated techniques are used to image and study blood vessels. This study uses the technique known as velocity encoded Phase Contrast MRA; other techniques such as black blood angiography (Jara & Barish 1999), contrast enhanced MRA (Roditi 2002), time-of-flight (Kim & Parker 2012), and steady-state free precession (Bieri & Scheffler 2013) are beyond the scope of this project.

PC-MRA is a versatile technique of producing contrast due to the movement of spins, most commonly in blood, through the gradient fields. Phase shifts are introduced and removed within a slice by the application of bipolar gradients. In this way stationary spins will be completely refocused while moving spins will encounter a non-matched refocusing pulse due to their altered location, and are therefore left with a net phase accumulation. The phase change experienced by each moving spin is proportional to its velocity in the direction that the bipolar gradient was applied. The phase change is also dependent on the timing between the gradient lobes and is discussed further in Section 2.2.2. In general, PC-MRA acquisitions are based on gradient echo sequences (Markl & Leupold 2012). Often RF-spoiling is used to reduce acquisition time for both Two-dimensional (2D) and Three-dimensional (3D) scans (Markl et al. 2012, 2011).

Gradient Echo Sequences

Gradient Echo (GRE) sequences are commonly used in PC-MRA due to their fast acquisition times which make them ideal for 3D and even "Four-dimensional (4D)", or time resolved 3D, imaging (Markl et al. 2012). As can be seen in Figure 2.3, GRE uses magnetic gradients to initially dephase and then refocus spins to form the echo during the frequency encode, or readout, phase, as opposed to using a 180° RF refocusing pulse as is the case in spin echo sequences. The prolonged application of gradient fields causes a rapid loss of phase coherence between the spins. Upon application of a gradient of opposite polarity the spins will begin to rephase and form an echo signal. Basic GRE sequences will use $\{\mathrm{TR} \mid \mathrm{T}_1 \gtrapprox \mathrm{TR} \gg \mathrm{T}_2\}$ and commonly uses $\alpha < 90^\circ$ to produce contrast between tissues (Markl & Leupold 2012).

Fast GRE pulse sequences use $\{\mathrm{TR} \mid \mathrm{TR} < \mathrm{T}_2, \mathrm{TR} \ll \mathrm{T}_1\}$ and are of the most interest

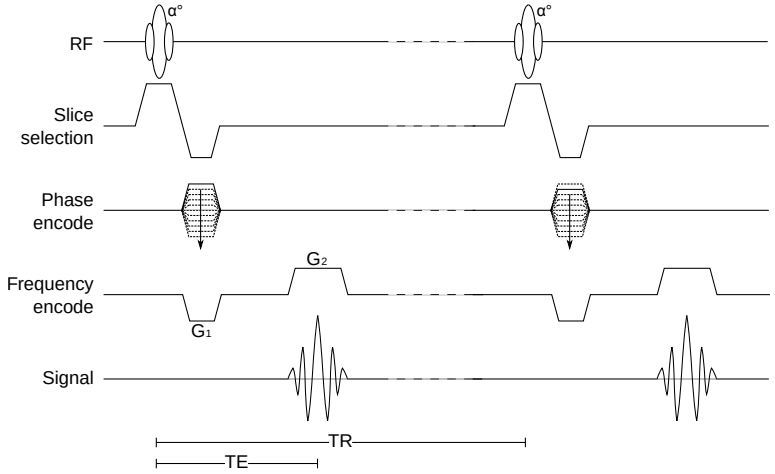


Figure 2.3: An example of a typical GRE pulse sequence. G_1 and G_2 are the amplitude of the dephasing and readout lobe, respectively.

to this project as they allow for greatly reduced acquisition times. Similar to basic GRE sequences, the flip angle is very important for determining the contrast between tissues and is below 90° (Markl & Leupold 2012). Both the TE and TR are on the order of milliseconds (Bernstein et al. 2004, chap. 14).

For a normal gradient pulse the amount of phase accumulated (in radians) from the point at which the excitation pulse is applied, at t_0, to receiving the signal, at TE, is described by Equation (2.2).

$$\begin{aligned}\phi(\mathbf{r}, \mathrm{TE}) - \phi(\mathbf{r}, t_0) &= \int_{t_0}^{\mathrm{TE}} \omega_0(\mathbf{r}, t)\mathrm{d}t \\ &= \gamma\, \Delta B_0(\mathrm{TE} - t_0) + \gamma \int_{t_0}^{\mathrm{TE}} \mathbf{G}(t)\mathbf{r}(t)\mathrm{d}t \\ \phi(\mathbf{r}, \mathrm{TE}) &= \phi(\mathbf{r}, t_0) + \gamma\, \Delta B_0(\mathrm{TE} - t_0) + \gamma \int_{t_0}^{\mathrm{TE}} \mathbf{G}(t)\mathbf{r}(t)\mathrm{d}t \end{aligned} \tag{2.2}$$

Where $\phi(\mathbf{r}, t_0)$ is a constant background phase value that results from inhomogeneities in the magnetic field and the initial phase; $\mathbf{r}$ is the spatial location of the spins; ΔB_0 is the local field inhomogeneity; $\mathbf{G}$ is the spatially and temporally varying local field gradient. γ is the gyromagnetic ratio. Equation (2.2) can then be expanded into its Taylor series equivalent given by Equation (2.3).

$$\begin{aligned}\phi(\mathbf{r}, \mathrm{TE}) &= \phi_0 + \gamma\,\mathbf{r_0}\int_0^{\mathrm{TE}} \mathbf{G}(t)\mathrm{d}t + \gamma\,\mathbf{v}\int_0^{\mathrm{TE}} t\,\mathbf{G}(t)\mathrm{d}t + \cdots + \gamma\,\frac{\mathbf{r}^{(n)}}{n!}\int_0^{\mathrm{TE}} t^n\,\mathbf{G}(t)\mathrm{d}t + \cdots \\ &= \phi_0 + \gamma\,\mathbf{r_0}\,\mathrm{m}_0(t) + \gamma\,\mathbf{v}\,\mathrm{m}_1(t) + \cdots + \frac{\gamma\,\mathbf{r}^{(n)}}{n!}\mathrm{m}_n(t) + \cdots \end{aligned} \tag{2.3}$$

Equation (2.3) gives the net phase as a Taylor sum of the background phase term and the phase accumulation due to the effect of n^{th} order gradient moments, m_n. These gradient moments influence the phase accrued by their respective n^{th} derivative of the temporally varying $\mathbf{r}$. γ is the gyromagnetic ratio. For the simple case of a stationary spin we can ignore all but the effects from the zeroth gradient moment, m_0, referred to as ϕ_{G}.

$$\phi_{\mathrm{G}}(\mathbf{r}, \mathrm{TE}) = \gamma\,\mathbf{r_0}\int_0^{\mathrm{TE}} \mathbf{G}(t)\mathrm{d}t \tag{2.4}$$

The formation of the gradient echo relies on all of the spins in the excited slice to rephase at TE which leads to the general condition for m_0:

$$\mathrm{m}_0 \equiv \int \mathbf{G}(t)\mathrm{d}t = 0 \tag{2.5}$$

For the condition in Equation (2.5) to be met in Figure 2.3 the areas under the frequency encode lobes must cancel out at TE. In general the second lobe is twice as long as the first which means the amplitudes, G_1 and G_2, are equal and given by the value G.

RF spoiling To produce T_1-weighted images with a very short TR it is necessary to dephase the spins, thus spoiling any remaining transverse magnetisation, before each excitation pulse. This can be achieved by varying the phase of each subsequent excitation pulse. It is for this reason that the receive and transmit coil must be phase-locked (Markl & Leupold 2012). RF spoiled GRE pulse sequences have been successfully implemented in 3D PC-MRA acquisitions (Markl et al. 2012).

Effects of Blood Flow

This project is focussed specifically on the imaging of blood flow in both biological and synthetic blood vessels. It is therefore important to understand the effect of spins moving through the magnetic field, as only stationary spins have been considered so far. Where it is assumed that stationary spins will encounter the same magnetic gradients during gradient reversal, moving spins will encounter a varying magnetic field, leaving it with a net phase accumulation. Ghost and flow artefacts are common imaging artefacts that are caused by motion and may occur as a result of periodic blood flow (Brown et al. 2014, chap. 23). Ghost artefacts are displaced duplications of the image while flow artefacts specifically refer to ghost artefacts caused by the movement of body fluids.
To compensate for the effect of velocity it is necessary to cancel the phase accumulation from the zeroth and first moments, m_1, from Equation (2.3).

$$m_1 \equiv \int t\,\mathbf{G}\mathrm{d}t = 0 \tag{2.6}$$

An extra bipolar gradient can be added to the standard GRE sequence on any combination of encode gradients to meet this condition. This creates a three lobed design, shown in Figure 2.4. The amplitude and timing of the sequence can be derived by solving Equations (2.5) and (2.6). The second lobe will have the opposite polarity to the first and readout lobes. This sequence is then able to null both stationary spins and spins moving with uniform velocity in the direction of the applied velocity compensation gradient (Brown et al. 2014, chap. 23).

It is often the case, especially for PC-MRA, that one would not want to compensate for velocity but instead would want to be able to quantify these flow effects.

Flow encoding gradients Flow Encoding (FE) gradients are used to encode information about the motion of spins through the gradient fields. These gradients are employed for imaging macroscopic flow, such as blood flow and cerebrospinal fluid. Flow encoding gradients can be applied to any combination of axes which allows for the quantification of flow in any arbitrary direction. The most common flow encoding gradient, and implemented in the pulse sequence used in this project, is the velocity encoding gradient. The velocity encoding gradient is a bipolar gradient. This means it is comprised of two

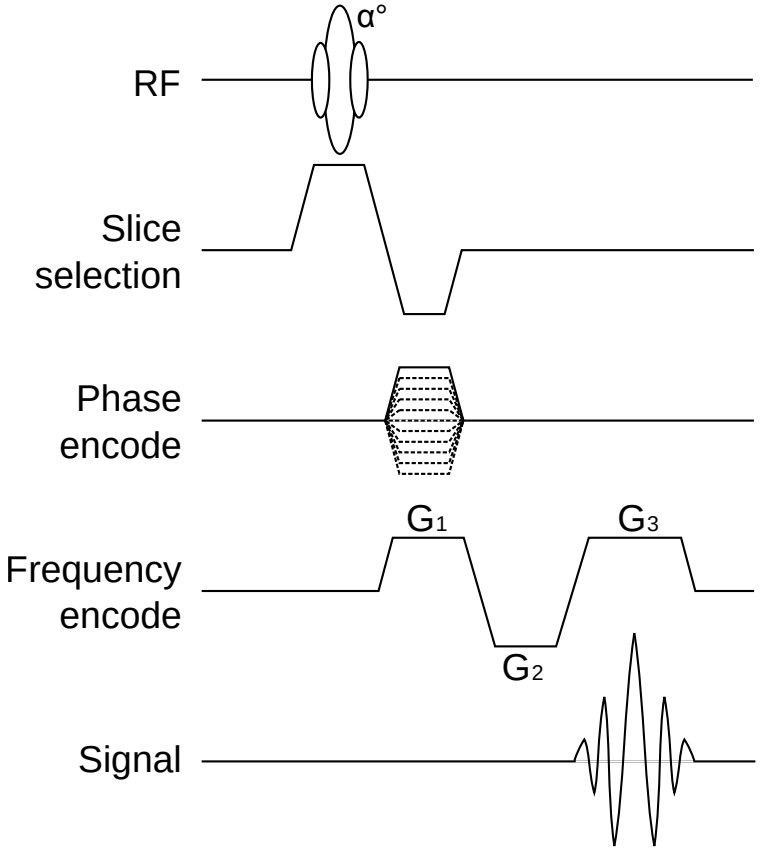


Figure 2.4: An example of a typical GRE pulse sequence with velocity compensation on the frequency encode axis. The amplitudes of the first second lobe, G_1 and G_2, can vary with respect to the readout amplitude, G_3.

lobes of opposite polarity but each lobe has an equal area so that the cumulative area of the gradient is zero. The effect of this is that there will be no net phase accumulation for stationary spins. Whereas spins in motion will encounter a changing magnetic gradient which leaves them with a net phase accumulation directly proportional to their velocity (Bernstein et al. 2004, chap. 9).

2.2.2 Velocity Encoding

Velocity encoded PC-MRA is achieved by the application of a bipolar FE gradient, in conjunction with a fast GRE pulse sequence on one, or any combination, of the gradient axes. It is most common that the FE gradient is applied to each gradient axis individually but it is also possible to apply it to any combination of the axes to encode velocity in an arbitrary direction. A typical velocity encoding pulse can be seen in Figure 2.5. The bipolar gradients can be joined with the other gradient lobes in an effort to reduce both the TR and TE of the sequence (Bernstein et al. 2004, chap. 9). Although this also

leads to a three lobed design, as with the velocity compensating technique, it is distinct in that it only fulfils the stipulation of Equation (2.5).

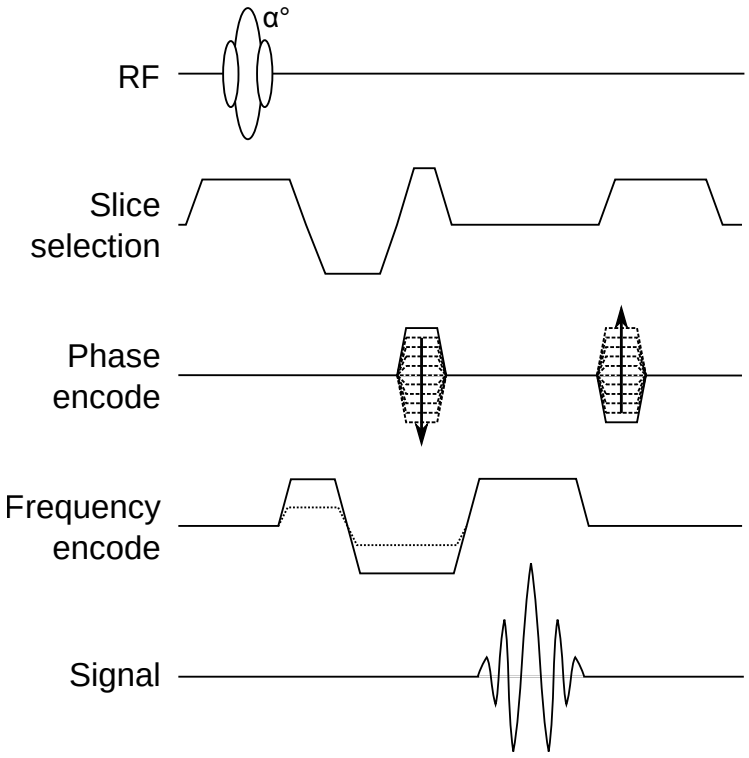


Figure 2.5: An example of a fast GRE pulse sequence with velocity encoding. The solid lines show the flow compensated gradients while the dotted lines show the flow encoding gradients for the frequency encoding axis. Adapted from Bernstein et al. 2004.

It can be assumed that the velocity is constant if it varies slowly enough compared to the temporal resolution which allows higher order terms to be ignored. By assuming a constant velocity it is only necessary to consider the first two terms, the zeroth and first order gradient moments. This means that the phase accumulation experienced by each spin is dependent on its position, $\mathbf{r_0}$, and its velocity, $\mathbf{v}$. The application of the bipolar gradients effectively nulls the phase contribution from stationary spins, as they encounter a negative and positive gradient lobe in the same position, leaving only phase from moving spins and the unknown background phase term. This background phase cannot be refocused by the bipolar gradient and has to be removed by subtracting two phase images acquired with opposite bipolar gradient polarities from one another (Markl 2005). Equation (2.3) can then be rewritten as Equation (2.7).

$$\Delta\phi = \gamma\, \Delta \mathrm{m}_1\, \mathbf{v} \tag{2.7}$$

The absolute velocity value at which the phase accumulation is equal to π is known as either the aliasing velocity or the Velocity Encoding Constant (v_{enc}) defined in Equation (2.8). This parameter is set by the MRI operator such that $\pm v_{\text{enc}}$ is the highest and lowest velocity value that can be uniquely encoded without causing aliasing, respectively. The effects of aliasing, or phase wrapping, are discussed in Section 2.2.3.

$$v_{\text{enc}} = \frac{\pi}{\gamma \left|\Delta \text{m}_1\right|} \tag{2.8}$$

$$\Delta \text{m}_1 = AT \tag{2.9}$$

From Equation (2.8) it can be seen that v_{enc} is controlled by the first gradient moment, defined by Equation (2.9) as the product of the area of a gradient lobe, A, and the time between the application of the gradient lobes, T. Substituting Equation (2.8) into Equation (2.7) and rearranging gives Equation (2.10).

$$\mathbf{v} = \frac{\Delta\phi}{\pi} v_{\text{enc}} \tag{2.10}$$

This equation gives a direct relationship between the phase difference of a certain spin and its velocity and that velocity will be uniquely detectable, provided the v_{enc} parameter is set correctly. The laminar nature of flow in blood vessels means flow varies spatially from a maximum velocity in the centre of the vessel to approximately zero flow along the vessel's lumen. Therefore it is useful to keep the spatial resolution of the reconstructed phase image high enough so that the velocity in a given voxel can be assumed to be constant (Brown et al. 2014, chap. 24). Figure 2.6 shows a side and end view of a straight tube, representing a blood vessel, and the characteristic parabolic flow profile that is indicative of laminar flow.

It is important to choose an appropriate value for the v_{enc} parameter as it sets the upper and lower bounds on velocities that can be uniquely represented. The choice of v_{enc} directly affects the sensitivity and specificity of the acquisition to flow. A low v_{enc} setting will be highly sensitive to low flow values but it is more likely to introduce phase wrapping while a high v_{enc} setting will likely avoid problems with phase wrapping but is less sensitive to flow overall (Bernstein et al. 2004). It is for this reason that the

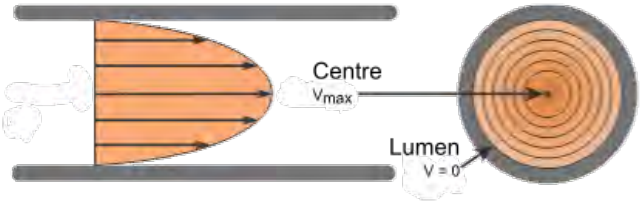


Figure 2.6: Side and end view of a typical laminar flow profile in a blood vessel. Adapted from CVPhysiology.com

v_{enc} value can not be set arbitrarily high to capture all possible speeds as there is a direct relationship between the noise in the image δ and the v_{enc} parameter shown in Equation (2.11).

$$\delta \propto \frac{v_{\text{enc}}}{\text{SNR}} \tag{2.11}$$

Due to the parabolic flow profile of blood, shown in Figure 2.6, all velocities from zero to the peak central flow velocity can be assumed to be represented within the vessel (Brown et al. 2014, chap. 24). For this reason it is generally accepted that v_{enc} should be set to the highest anticipated velocity in the area, this is commonly the peak velocity of a major artery (Markl 2005). Alternatively the v_{enc} parameter can be lowered further so that it is lower than the highest anticipated velocity. This will improve the Signal to Noise Ratio (SNR) of the image however it will also have the effect of adding aliasing, otherwise known as phase wrapping, in areas with high velocity flow (Brown et al. 2014, chap. 24).

2.2.3 Phase Wrapping

Phase wrapping occurs when movement is encountered that has a higher absolute velocity than the chosen v_{enc} parameter. If the velocity of a spin, v, can be described by $(2n-1)\cdot v_{\text{enc}} \leq v \leq (2n+1)\cdot v_{\text{enc}}$ then its phase, $\Delta\phi$, will be represented by $-\pi \leq \Delta\phi \leq \pi$ for $n \in \mathbb{Z}$. The value of n changes as the velocity reaches and surpasses either the upper or lower v_{enc} limit, this means n also represents the number of wraps encountered from a known phase reference point. The effects of phase wrapping on a linearly varying phase

gradient are shown in Figures 2.7a and 2.7b. The boundary between black and white pixels indicates a phase wrap and corresponds to a 2π discontinuity.

(a) The original phase map showing a linearly, spatially varying gradient with values from π (black) to 10π (white).

(b) The wrapped phase map introducing inconsistencies at odd integers of π. Black and white represent $-\pi$ and π respectively.

Figure 2.7: Original and wrapped phase maps

Phase Unwrapping

The process for removing wrapping artefacts from phase images is simple in theory. A pixel is chosen as a reference point in the image, the image is then unwrapped by either adding or subtracting 2π offsets based on how many phase discontinuities have been crossed between the reference and the current pixel. If the boundary crossed changes from white to black, π to $-\pi$, the offset should be added whereas if the change is from black to white, $-\pi$ to π, the offset should be subtracted. In theory, this will lead to a path independent solution; this is because no matter which path is chosen between two pixels, the net phase offset that is added will be the same. This is illustrated below in Figure 2.8.

Although the longer path crosses more boundaries it must double back on itself, effectively cancelling the effect of those crossed boundaries, to reach the point at B.
The preceding description of phase unwrapping has only considered a situation where the reconstructed phase image has been completely noise free, and high enough resolution that there is always a sufficient amount of space between wrapping boundaries. This is rarely the case in practice; this is due to the fact that there is noise inherent to the scanner such as B_0 and gradient field inhomogeneities, context specific noise such as low SNR, flow voids or ghosting from patient movement, as well as fluctuations in

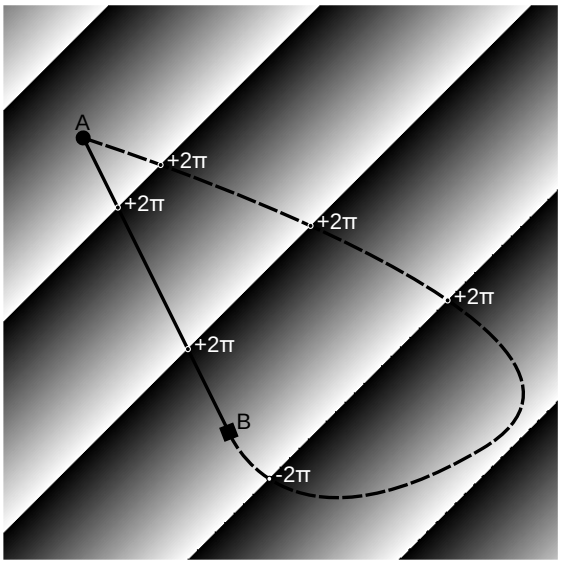


Figure 2.8: Although the two paths between A and B differ in length and the number of inconsistencies encountered, the net phase offset will not change.

the spins' magnetism due to random thermal effects. The noise effects are then coupled with the fact that the spatial resolution will be chosen to reduce scan times and as such may be low.

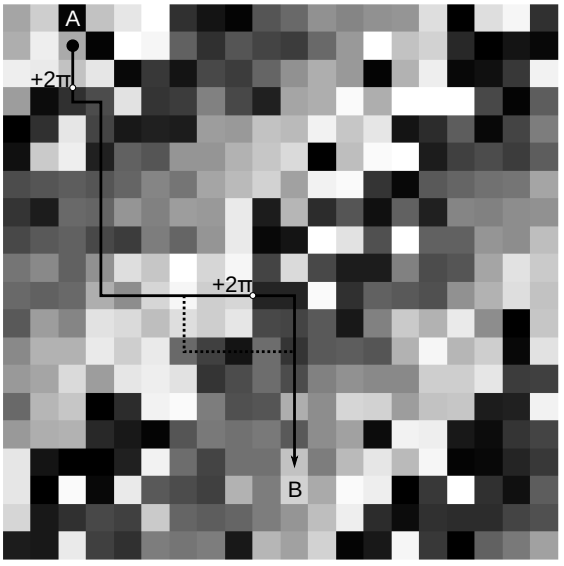


Figure 2.9: A more accurate representation of a wrapped phase map with random noise and a lower resolution.

Figure 2.9 shows the effects of a lower resolution and random noise on the phase image. Of particular interest in Figure 2.9 is that the map can no longer be considered path independent as the noise has created several areas on the discontinuity boundaries that no longer cause a 2π discontinuity. This is demonstrated by the solid path from **A** to **B**. It initially encounters a discontinuity but after its path diverges, the dashed path

doesn't encounter a second discontinuity while the solid path does. Each path will thus give a different result at **B**.

Techniques for performing phase unwrapping as well as methods for avoiding these sources of inconsistency and maintaining path independence are detailed in Chapter 4.

Chapter 3

Acquisition and Pre-processing

3.1 Materials and Techniques

MRI data was collected from the upper arm of healthy volunteers and patients receiving haemodialysis treatment. The scanning protocol used for both the volunteers and patients included velocity encoded 2D and 4D PC-MRA sequences (Markl et al. 2012), these sequences were used to directly image blood flow. All of the protocols used were approved by the Faculty's Human Research Ethics Committee.

Two sets of images are produced for each PC-MRA scan; a magnitude and a phase set. The phase set contains a number of subsets of images based on the number of velocity encoding directions used. The magnitude set of images shows the gross anatomy of the arm, most importantly the position of the arterial and venous vessels. The set of phase images contains velocity encoded phase values proportional to the flow velocity in a given voxel. The sign of the phase value indicates directionality with respect to the velocity encoding direction.

The images gathered from the MRI acquisition were pre-processed to remove noise from the phase images and inhomogeneities inherent in the scanning sequence.

The phase images may contain wrapping artefacts due to high velocity blood flow which must be corrected to accurately process and capture information about the volunteer or patient's blood flow. The phase images were manually inspected during the pre-processing phase to determine exactly which datasets had been affected by phase wrapping. An additional step of phase unwrapping was then applied to these datasets. The methods applied during the phase unwrapping stage are described later in Chapter 4.

Finally, a tool for visualising and quantifying flow was used to extract various characteristics about the blood flow and the vessels. This information was visually inspected to determine if a dataset was not corrupt and if the phase unwrapping operations had been applied successfully.

3.1.1 Available Resources

The following resources were used during the course of this study:

- 1.5T Full-body MRI scanner (MAGNETOM Symphony, Siemens AG, Erlangen)
  - Flexible arm coil
  - ECG monitor
- MATLAB (Mathworks, Natick, MA) on an Intel Core i7 (16GB RAM)
  - Image processing toolbox
  - Previously developed pre-processing and visualisation GUI and code (Uppman 2009)
  - Super tool v026 2012a win64 (Department of Clinical Radiology, University of Freiburg, Freiburg im Breisgau)
    - *"Velomap" tool* (Bock et al. 2007)
    - *"flow_tool" v3.0* (Stalder et al. 2008)

3.2 Imaging Protocol

The 1.5T MRI scanner (MAGNETOM Symphony, Siemens AG, Erlangen) was used to collect data from the upper arm of five healthy volunteers (Age: 27.2 ± 3.82 years) and eight patients (Age: 36.5 ± 11.6 years), of which five patients had brachiocephalic fistulas (Age: 35 ± 14.7 years) and three patients had brachiocephalic grafts (Age: 39 ± 9.17 years). 2D and 3D acquisition techniques were based on the 3D MRI velocity mapping package developed by Markl et al. (2012) for time resolved 2D and 3D PC-MRA. The 2D acquisitions were used to quickly determine the blood velocity profiles at proximal, distal and intermediate sites within the patients' arms. The time resolved 3D, henceforth

known simply as 4D, acquisition was used to determine both the structure of the blood vessels and the blood velocity profile throughout the upper arm. All volunteers and patients were scanned with both imaging protocols.

3.2.1 2D PC-MRA

The 2D PC-MRA scans were performed in the axial plane and used velocity encoding in the axial, or z-, direction. The scanning parameters used to scan the volunteers and patients are given in Table 3.1. Volunteer scans were used to optimise and test parameters for the patient scans.

Table 3.1: Details of 2D PC-MRA scans for volunteers and patients

Patient ID	v_{enc} **[cm/s]**	**№of Images**	**Pixel Spacing [mm/mm]**	**TR [ms]**	**TE [ms]**	**Flip Angle [°]**
Volunteer 1	15	12	0.70/0.70	84.0	4.17	15
	30	22	0.70/0.70	49.6	3.41	
	40	22	0.94/0.94	46.4	3.17	
	60	22	0.94/0.94	44.0	2.89	
Volunteer 2	20	9	0.70/0.70	79.2	3.82	15
	80	11		66.0	2.73	
Volunteer 3	20	9	0.70/0.70	79.2	3.82	15
	80	12		66.0	2.73	
Volunteer 4	80	12	0.70/0.70	66.0	2.73	15
	80	28		22.8	2.73	
Volunteer 5	20	9	0.70/0.70	79.2	3.82	15
	80	12		66.0	2.73	
Fistula 1	20	12	0.70/0.70	79.2	3.82	15
	80	14		66.0	2.73	
Fistula 2	20	12	0.70/0.70	79.2	3.82	15
	80	14		66.0	2.73	
Fistula 3	80	14	0.70/0.70	66.0	2.73	15
Fistula 4	80	14	0.70/0.70	66.0	2.73	15
Fistula 5	80	57	0.70/0.70	11.0	2.73	15
	150	66		10.4	2.46	
	350	28		24.0	3.23	
Graft 1	20	12	0.70/0.70	79.2	3.82	15
	80	14		66.0	2.73	
Graft 2	80	10	0.70/0.70	66.0	2.73	15
Graft 3	20	12	0.70/0.70	79.2	3.82	15
	80	14		66.0	2.73	

Two slices were chosen at proximal and distal sites in the upper arm of both patients and volunteers; a third slice was also chosen at the level of the anastomosis of each patient. Several images were acquired during the course of several cardiac cycles at different time points, this produces a set of 2D magnitude and phase images over one cardiac cycle. These scans can be performed in approximately one minute.

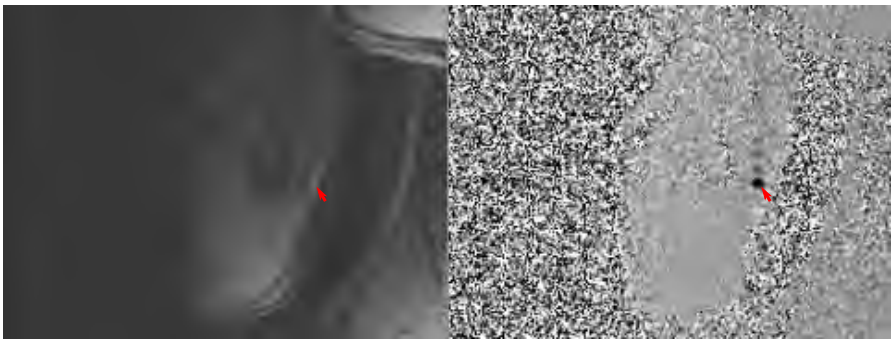


Figure 3.1: 2D PC-MRA magnitude and phase images; the red arrows show the location of the brachial artery. Flow from head to toe appears darker in the phase image.

3.2.2 4D PC-MRA

The 4D acquisition was performed in the sagittal plane and used velocity encoding in three orthogonal directions, that is the x-, y-, and z-directions. Table 3.2 gives the parameters used for each patient scan.

Table 3.2: Details of 4D PC-MRA scans for volunteers and patients

Patient ID	v_{enc} **[cm/s]**	**№of Images**	**№of Slices**	**Pixel Spacing [mm/mm]**	**Slice Thickness [mm]**	**TR [ms]**	**TE [ms]**	**Flip Angle [°]**
Volunteer 1	60	20	20	1.72/1.72	2	45.6	3.10	7
Volunteer 2	20 90	14	20	1.56/1.56	2	56.0 52.8	4.38 3.89	7
Volunteer 3	20 90	28	16	1.56/1.56	2	27.2 26.4	4.18 3.89	7
Volunteer 4	90	26	16	1.56/1.56	2	26.4	3.89	7
Volunteer 5	20	28	16	1.56/1.56	2	27.2	4.18	7
Fistula 1	90	28	30	1.56/1.56	2	26.0	3.89	7
Fistula 2	20	28	16	1.56/1.56	2	27.2	4.18	7
Fistula 3	90	28	30	1.56/1.56	2	26.0	3.89	7
Fistula 4	90	28	40	1.56/1.56	2	26.4	3.89	7
Fistula 5	250	12	32	1.56/1.56	2	11.0	2.73	7
Graft 1	20	28	26	1.56/1.56	2	27.2	4.18	7
Graft 2	90	28	24	1.56/1.56	2	26.0	3.89	7
Graft 3	20	28	16	1.56/1.56	2	27.2	4.17	7

The imaged 3D volume was positioned in the volunteers' upper arm so that both the brachial artery and cephalic vein could be measured in one acquisition. For the patients, the volume was chosen to cover the brachial artery, the cephalic vein or graft, the anastomosis of the artery and vein or graft, and the brachial artery distal to the anastomosis. Similar to the 2D PC-MRA acquisition, a slice is imaged multiple times to produce a set of images throughout one cardiac cycle, however, it differs in that several slices are acquired throughout the volume. This leads to greatly increased scan times, generally between 20 and 30 minutes.

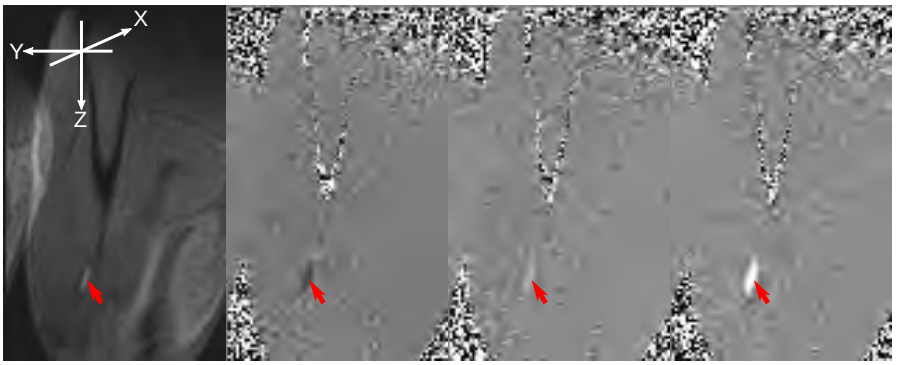


Figure 3.2: 4D PC-MRA magnitude and phase images; the red arrows indicate a section of the brachial artery. Flow in the direction of the x-, y-, and z-axes will be brightest in the phase image for spins moving from right to left, front to back, and head to toe, respectively. In order: magnitude, x-, y-, and z-axis velocity encoded phase.

3.3 Pre-processing

The image pre-processing was performed using a software package called "Velomap". A pre-processing tool written in MATLAB (Mathworks, Natick, MA) and developed by Bock et al. (2007) from the Department of Diagnostic Radiology, University of Freiburg, Germany. A modified version of a software package developed by Uppman (2009) was also used.

Several sources of noise exist that commonly affect PC-MRA by introducing phase offset errors; such as eddy current effects (Lankhaar et al. 2005, Lorenz et al. 2014) and mismatched encoding velocity (Lotz et al. 2002) as well as small errors caused by local field and gradient inhomogeneities (Lotz et al. 2002) that are found in all MRI scans. Both software packages attempt to remove these errors by the application of various filtering techniques and other corrections in order to process the MR images into a homogeneous form which can then be used to perform further processing and analysis of the blood flow.

The method for noise masking used by Bock et al. (2007) and Uppman (2009) was to perform a combination of simple magnitude thresholding and a pixelwise comparison of standard deviations along the temporal axis. Areas of high noise are commonly found in areas of low signal and so a simple magnitude threshold is able to remove the majority of noise. Compared with static tissue and flow, noise will vary quickly along the time axis and therefore have a far greater standard deviation which can be used as a second threshold to remove any remaining noise. There is a caveat to this however, Lotz et al. (2002) states that it is possible that this technique may introduce additional

phase offsets and so should be used sparingly, specifically in cases of high noise or where large phase offsets are already present. Additionally, Uppman (2009) included a masking function so the user could manually define a Region of Interest (ROI) in order to decrease pre-processing time.

The gradients used in these PC-MRA scans can also be detrimental to the quality of the image, this is due to the time-varying gradients setting up changing magnetic fields which in turn induce currents within the magnet and coils. These induced currents, known as eddy currents, then create further unwanted magnetic fields (Bernstein et al. 2004). Eddy currents are a major contributor to phase offset errors (Lankhaar et al. 2005). Eddy current correction was achieved by separating the flow signal from static tissue, fitting a plane to the static region, and finally subtracting the fitted plane from each frame (Bock et al. 2007). Static tissue shows little to no variation in time whereas flow, especially pulsatile flow, by its nature will vary in time. This means that the time dependent standard deviations can have a threshold applied to perform the tissue separation. A first or second order plane is then fitted to the static region using the least squares method and subtracted from each image at every time point.

The "flow_tool" is a software packaged developed in MATLAB (Mathworks, Natick, MA) by Stalder et al. (2008) from the Department of Clinical Radiology, University of Freiburg, Germany. The magnitude and phase data structures created by the "Velomap" tool were loaded into "flow_tool" for an initial visual inspection. The magnitude and phase images are used together to define a region of interest around a vessel, shown in Figure 3.3. After defining a region of interest the program is run and flow data from the region is returned. This data is inspected to determine if any phase wrapping has occurred. Figure 3.4 shows a vessel that has been affected by phase wrapping thus giving inaccurate blood flow values. Any datasets like this will be flagged as requiring phase unwrapping. During phase unwrapping it is important that the format of the data structure from the "Velomap" tool is maintained so that the unwrapped datasets can still be inspected using the "flow_tool".

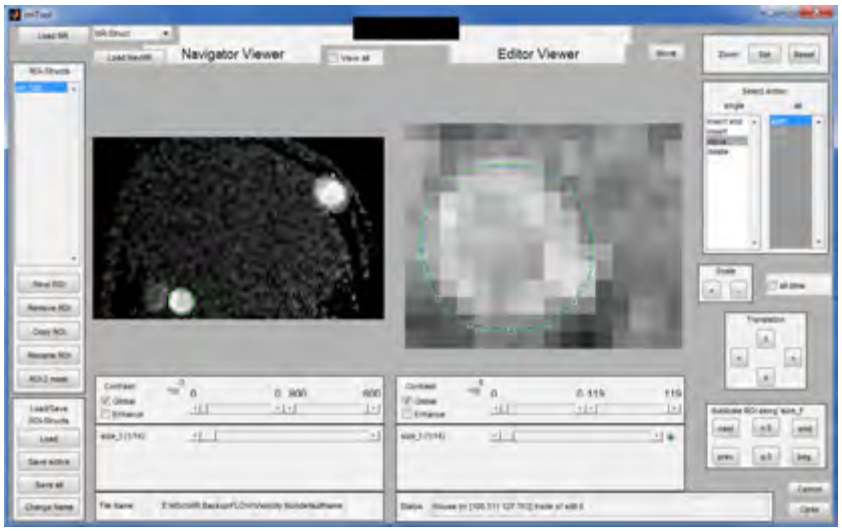


Figure 3.3: Defining a region of interest around the brachial artery of patient Fistula 4

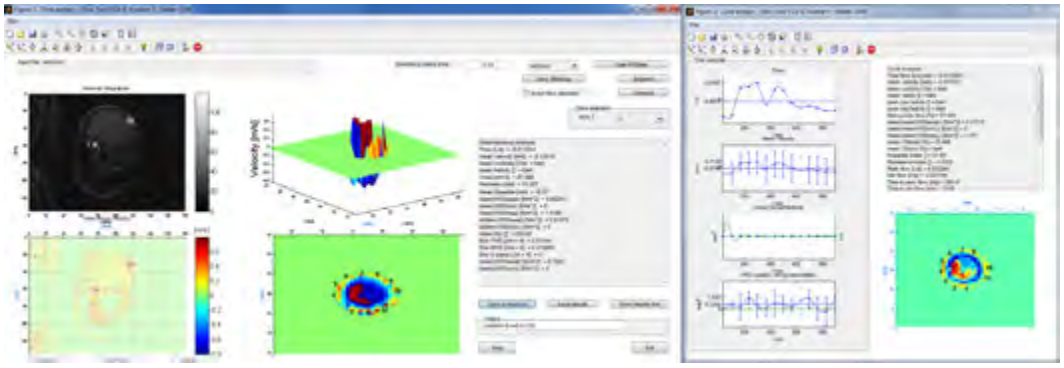
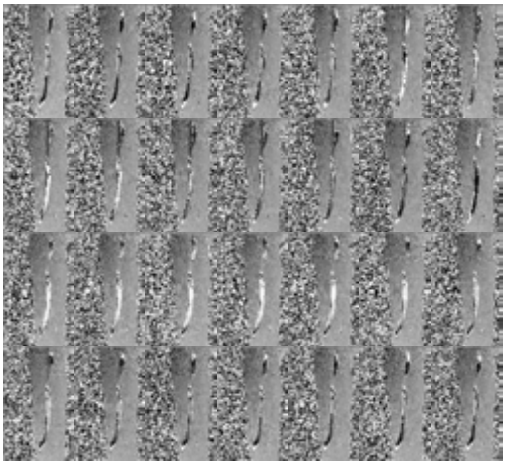


Figure 3.4: Example of phase wrapping during initial visual inspection of flow values

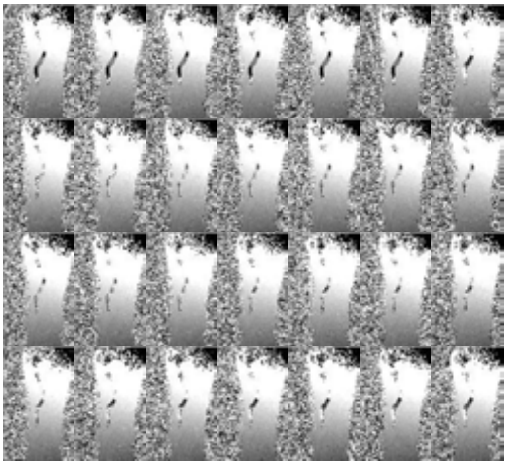
3.4 Data Acquired

The volunteer scans were used to test different sets of parameters which helped to determine what settings would be used during the imaging of the AV fistula and graft patients. Five volunteers were scanned at a high and low v_{enc} setting during the 2D PC-MRA acquisition to best capture the dynamics of arterial and venous flow respectively. The volunteers were initially scanned at a high and low v_{enc} setting during the 4D PC-MRA acquisition as well, however, the later volunteers were scanned at only one of the v_{enc} settings to reduce the scan time. The low v_{enc} parameters were chosen to determine whether the low velocity venous flow and high velocity arterial flow could be acquired during a single acquisition. Due to the sensitivity-specificity dichotomy, described in Section 2.2.2, it was known that these low v_{enc} scans would contain phase wrapping artefacts and as such Chapter 4 details several methods for performing phase unwrapping.

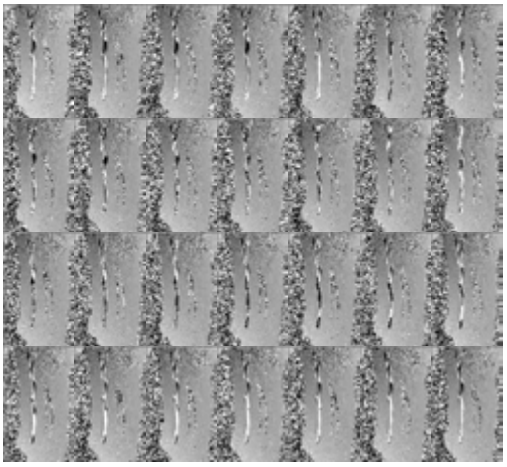
Three patients, Fistula 2, Graft 1, and Graft 3, were scanned at a v_{enc} setting of 20 cm/s during the 4D PC-MRA acquisition which was too low to yield any usable results as the phase wrapping artefacts were too severe as can be seen in Figure 3.5. Tests performed using DDU on subsequent fistula patients produced results that made it apparent that these scans performed with a low v_{enc} setting would not provide valid results. For this reason higher v_{enc} settings were used for subsequent patient acquisitions.



(a) All frames of a single slice from Fistula 2



(b) All frames of a single slice from Graft 1

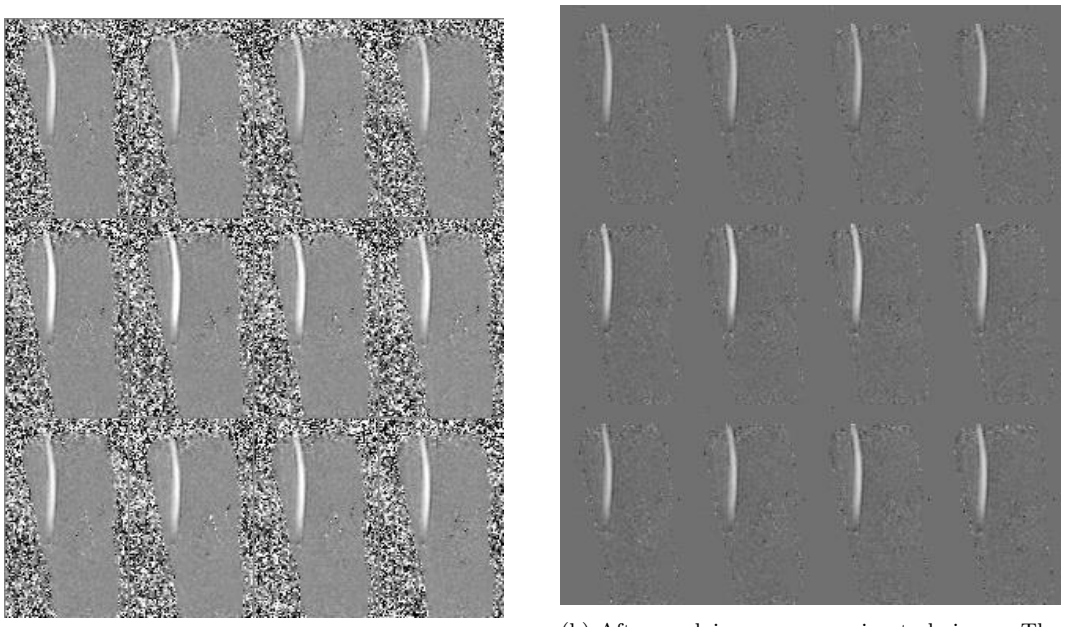


(c) All frames of a single slice from Graft 3

Figure 3.5: 4D PC-MRA from three patients acquired with a very low v_{enc} setting

Pre-processing was performed on the remaining volunteer and patient data to remove noise from the image and to remove any phase offsets that may have been present. All images were successfully pre-processed. One patient, Fistula 5, was scanned at much higher v_{enc} settings than the other patients. The v_{enc} settings were chosen to avoid phase wrapping as closely as possible in the fistula and in the surrounding vessels during both the 2D and 4D PC-MRA acquisitions. If there was any phase wrapping, the unwrapping procedure was trivial as the wrapped area would amount to a few pixels. This patient

was then used as a gold standard for validating the flow values of the other patients as well as determining if the unwrapping procedures were successful. One slice from the pre-processing of the patient Fistula 5 is shown below in Figure 3.6.



(a) Before applying pre-processing techniques

(b) After applying pre-processing techniques. The anatomy is now clearly defined

Figure 3.6: All frames from a single slice from the 4D PC-MRA acquisition of Fistula 5

In Figure 3.6b, the majority of background noise has been removed from Figure 3.6a which leaves only the anatomy of the patient's arm. The area of flow, in this case the brachial artery, is also now clearly defined.

Chapter 4

Phase Unwrapping

The basic theory of phase wrapping and unwrapping was previously discussed in Section 2.2.3. A more mathematical description of phase wrapping is given in Equations (4.1) and (4.2).

$$\begin{aligned} \phi_{x,y} &= \mathrm{W}[\Phi_{x,y}] \\ &= [(\Phi_{x,y} + \pi)\ mod\ 2\pi] - \pi \end{aligned} \tag{4.1}$$

And more generally,

$$\mathrm{W}[\Phi_{x,y}] = \Phi_{x,y} + 2n_{x,y}\pi \tag{4.2}$$

W is the wrapping operator that bounds the phase values such that $\mathrm{W}[\cdot] \in (-\pi, \pi]$, ϕ is the wrapped phase map reconstructed from the PC-MRA pulse sequence, Φ is the true phase map which contains no wrapping, *mod* is the modulo function, n is some integer value that fulfils the wrapping operators boundary requirements, and $[x, y]$ correspond to 2D pixel co-ordinates within the phase maps although these can be extended to cover 3D volumes and 4D time resolved volumes. All phase unwrapping methods attempt to find the integer value $n_{x,y}$ for each pixel.

In this work several path following phase unwrapping techniques were investigated and are detailed in Sections 4.1 to 4.4. Four methods of phase unwrapping were chosen to be

applied to the PC-MRA datasets that were affected by phase wrapping, namely: temporal, Goldstein branch cut, Phase Derivative Variance (PDV), and the 3D branch cut extension. The temporal phase unwrapping method was also used in conjunction with the Goldstein branch cut method. These methods were validated using simulated data with various noisy densities, see Section 4.5. The results of the unwrapping techniques are found in Chapter 5.

4.1 Temporal Phase Unwrapping

The method used by (Bock et al. 2007) is known as temporal phase unwrapping. The method is based on a One-dimensional (1D) unwrapping algorithm, see Section 4.2.1, however, instead of performing the unwrapping algorithm along a spatial dimension it is performed along the time axis (Huntley & Saldner 1993). This method uses a very simple approach to unwrap phase data based on the assumption that the TR of the PC-MRA scan has been kept short enough that the phase of corresponding pixels in contiguous time frames would not differ by more than 2π. It is also important that the boundaries do not vary from frame to frame.

Each pixel is unwrapped independently of one another which, along with the stipulation that boundaries must not vary between frames, means this method is spatially independent and can be applied to any n-dimensional scan provided there are two or more time frames. The spatial independence of the temporal phase unwrapping algorithm means that localised areas of low SNR will not adversely affect areas with higher SNR, which is likely good data (Huntley & Saldner 1993).

The position vector $\mathbf{r}$ will represent any n-dimensional spatial co-ordinate and $(t = 1, 2, \ldots, t_f)$ represents the time frame. The first step is to determine if a 2π phase discontinuity has occurred between frames.

$$\Delta\phi(\mathbf{r}, t) = \phi(\mathbf{r}, t) - \phi(\mathbf{r}, t-1) \tag{4.3}$$

$$\delta(\mathbf{r}, t) = \lfloor \frac{\Delta\phi(\mathbf{r}, t)}{2\pi} \rceil \tag{4.4}$$

$\Delta\phi(\mathbf{r}, t)$ is the phase difference between frame t and $t-1$, $\lfloor\cdot\rceil$ rounds the operand to the nearest integer. Next, the total number of phase discontinuities is summed for each frame.

$$n(\mathbf{r}, t_f) = \sum_{t=2}^{t_f} \delta(\mathbf{r}, t) \tag{4.5}$$

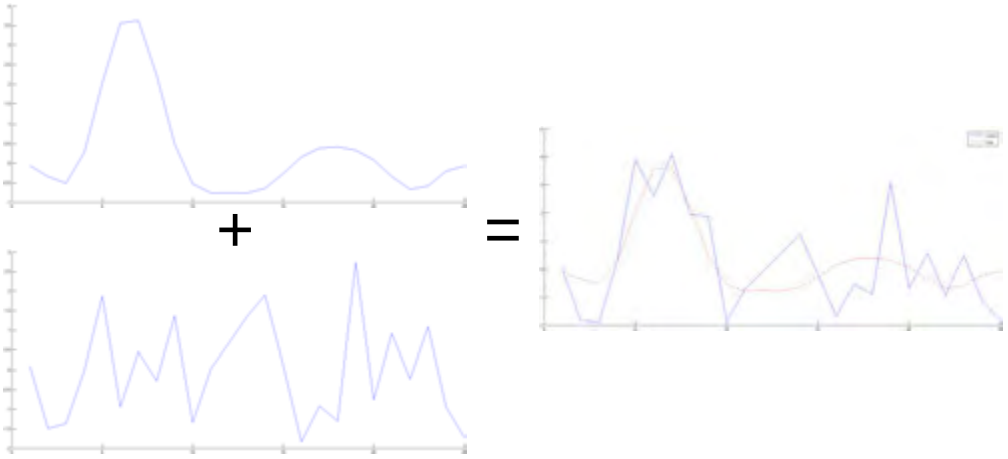
The value of $\delta(\mathbf{r}, 1)$ is set to zero as the first frame is chosen as a reference frame, meaning that these phase values are assumed to be the true unwrapped phase values. Therefore all phase discontinuity calculations are performed with reference to the first frame. Finally, $2\pi n(\mathbf{r}, t)$ is subtracted from its corresponding wrapped phase value.

$$\phi'(\mathbf{r}, t) = \phi(\mathbf{r}, t) - 2\pi n(\mathbf{r}, t) \tag{4.6}$$

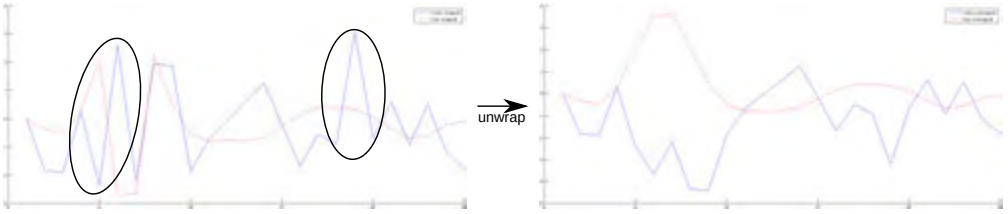
This final operation gives the value $\phi'(\mathbf{r}, t)$ which is the approximation of $\Phi(\mathbf{r}, t)$. If $\phi'(\mathbf{r}, t) = \Phi(\mathbf{r}, t)$ then the operation was successful and the pixel is said to be unwrapped. It is, however, difficult to determine this in practice as Φ is often not available. A further problem with working with only one dimension arises when there is noise along the unwrapping path. Noise by itself is not necessarily detrimental to a given unwrapping path as long as its value doesn't result in a phase discontinuity that would otherwise not exist. Conversely, noise can also mask a true phase discontinuity if the noisy phase value causes Equation (4.3) to be between $-\pi$ and π as shown in Figure 4.1. Clearly the unwrapped flow curve, ϕ', does not match the true flow curve, Φ, because the noise in the image has masked a true phase discontinuity.

4.2 Spatial Phase Unwrapping

This method of unwrapping phase spatially, similar to the temporal method, uses a path following technique to guide the unwrapping algorithm through the phase volume. In contrast to the temporal technique, these techniques attempt to remove phase inconsistencies along the spatial dimensions. In principle this is achieved by integrating the phase differences along all paths through the phase map. Provided that the stipulation



(a) A simulated blood flow curve has random noise added along the temporal axis. This gives the true phase curve, Φ.



(b) The wrapped curve, ϕ, is then unwrapped to give the curve, ϕ'. The circled area on the left shows where a true phase discontinuity was masked; the circled area on the right shows where a spurious phase discontinuity was introduced.

Figure 4.1: Example of how noise can cause the temporal phase unwrapping operation to fail. The dashed lines shows what the noisy curve would look like with no noise for each stage.

that the phase map has path independence is met, this method can be achieved by applying a flood fill algorithm to the phase map.

4.2.1 One-dimensional Phase Unwrapping

The simplest case of spatial phase unwrapping is analogous to temporal phase unwrapping in that one spatial axis is chosen and the unwrapping operator is applied to each reference pixel independently. Itoh (1982) describes the process for removing the 2π offsets from phase angles present in high resolution imaging of objects through atmospheric turbulence. ϕ and Φ are again the wrapped and unwrapped phase maps, respectively; $(m = 1, 2, \ldots, M)$ is the 1D position index chosen along an arbitrary spatial axis.

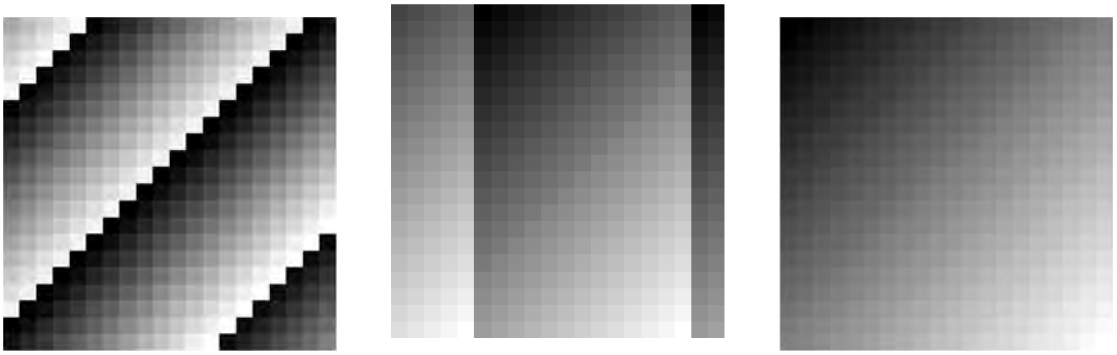
Equation (4.3) is slightly modified to apply to spatial dimensions:

$$\Delta\phi(m) = \phi(m) - \phi(m-1) \tag{4.7}$$

The phase gradient, $\Delta\phi$, is then wrapped and integrated:

$$\Phi(j) = \phi(1) + \sum_{m=2}^{j} W[\Delta\phi(m)] \tag{4.8}$$

The image is unwrapped along both its rows and columns, this will give the effect of unwrapping from a specific reference pixel. The reference pixel is by definition already unwrapped so one must take care to choose a pixel with a known phase value. For example, we will unwrap Figure 4.2a with respect to the pixel at (1,1), the top left corner. First the image is unwrapped along each column starting from the first row, Figure 4.2b. Only the phase value at (1,1) is known to be true so only the first column is considered to be unwrapped. Next the image is unwrapped along each of its rows starting from the now unwrapped first column. This leads to the correctly unwrapped image in Figure 4.2c.



(a) Wrapped phase map. Black and white represent $-\pi$ and π, respectively.

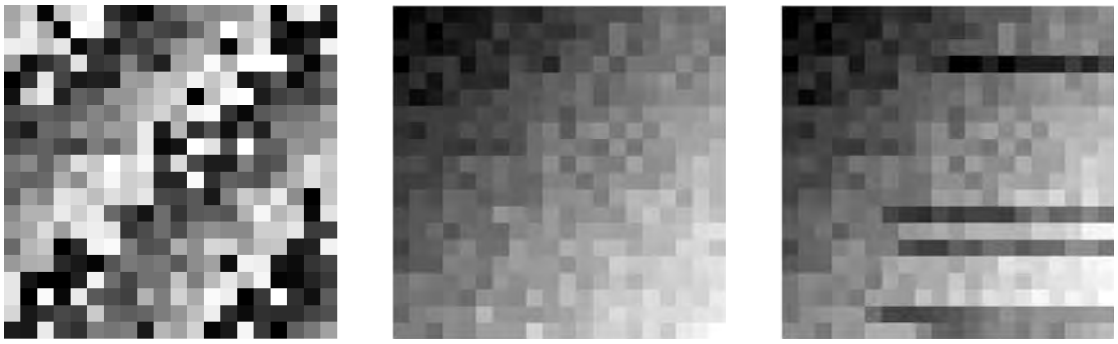
(b) During the unwrapping stage. Here the columns have been unwrapped; the rows are still wrapped.

(c) Unwrapped phase map. Black and white represent 0 and 6π, respectively.

Figure 4.2: The process of Itoh's 1D unwrapping algorithm as applied to a simple wrapped image

Itoh (1982) shows that the condition for Equation (4.8) to be met is $-\pi \leq \Delta\Phi < \pi$. That is the difference between two adjacent unwrapped phase values should be between $-\pi$ and π. In theory, provided the spatial resolution of the scan is adequate, this condition should be fulfilled. In practice, however, due to random noise in the image, this is often not the case. A random noisy pixel can introduce phase inconsistencies which may lead to an erroneous 2π offset throughout the array. The effect of these inconsistencies is shown in Figure 4.3.

Clearly these inconsistencies should be avoided during unwrapping as often as possible. The problem with Itoh's method is that in a 1D array there are no other directions



(a) Wrapped phase map with random noise.

(b) The true unwrapped phase map.

(c) Unwrapped phase map using Itoh's method.

Figure 4.3: The process of Itoh's 1D unwrapping algorithm as applied to a noisy wrapped image, column-wise and then row-wise

to divert the unwrapping algorithm along and so inconsistencies are almost guaranteed using this method.

4.2.2 Nearest Neighbour Algorithm

By extending Itoh's method to two spatial dimensions it is possible to overcome its limitations. In principle the method has not changed, a reference pixel is chosen and all pixels along a given path are unwrapped to give the true phase value at each point. The difference now being that there are multiple paths that can be taken and for the unwrapping algorithm to work consistently every path should return the same values for their respective pixels.

As discussed in the preceding 1D examples, noisy pixels add errors to the phase image which can lead to spurious 2π discontinuities throughout the image during the unwrapping stage. In two dimensions it is possible to identify these pixels and in doing so avoid them. An important characteristic of ideal wrapped phase maps, where phase wrap discontinuities are caused by phase wrapping only, is that the sum of the wrapped phase gradient, known as the net phase difference integral, in a closed loop is zero (Spottiswoode 2006). What can be inferred from this is that if the result of the integral is non-zero the loop includes at least one residue. In general a localised discontinuity source will create a pair of oppositely charged residues unless the discontinuity is located near the image boundary(Huntley 2001). To identify these residues the net phase difference integral is applied to each group of pixels in a 2×2 matrix. In Equation (4.9) a pixel, at position (x,y) in the image, and its three surrounding neighbours are investigated.

$$s(x,y) = \lfloor \frac{\phi(x,y) - \phi(x,y+1)}{2\pi} \rceil + \lfloor \frac{\phi(x,y+1) - \phi(x+1,y+1)}{2\pi} \rceil + \ldots \\ \lfloor \frac{\phi(x+1,y+1) - \phi(x+1,y)}{2\pi} \rceil + \lfloor \frac{\phi(x+1,y) - \phi(x,y)}{2\pi} \rceil \tag{4.9}$$

The possible values of $s(x,y)$ are 0 and ± 1. If $s(x,y)$ is non-zero, the pixel at position (x,y) is chosen to represent the residue even though it actually exists between the pixels. A positively charged residue has an s value of $+1$ and a negatively charged residue has a value of -1. The total phase accumulated by the net phase difference integral is determined by Equation (4.10).

$$S = 2\pi \sum_n s(x_n, y_n) \tag{4.10}$$

An enlarged excerpt from Figure 4.3a is shown in Figure 4.4. A localised discontinuity source, and its associated positive and negative residues, are depicted by the closed loops Figure 4.4 (a) and (b), respectively. In (c) the simplest case for these closed loops is shown, it contains no wrapping of any kind and thus there is no net phase accumulation. In (d) the loop encircles a 2π discontinuity caused by phase wrapping only and so there is no net phase accumulation.

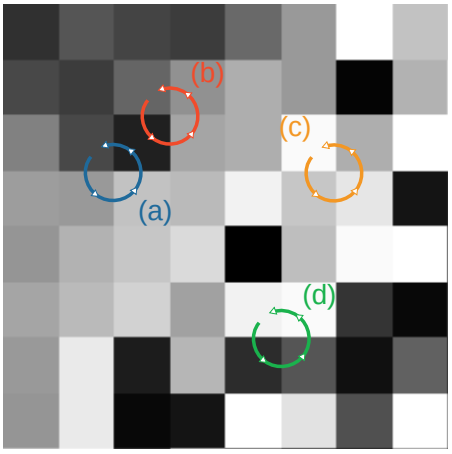


Figure 4.4: Types of closed loop net phase difference integrals. Clockwise from left: (a) and (b) Positive and negative phase residues, respectively. (c) No wrapping and no residues. (d) Wrapping is present but no residues. Black and white represent $-\pi$ and π, respectively.

A path that encircles just one residue, or if the sum of residues is non-zero, will have a net phase accumulation, however, if the path were to encircle an equal number of

positively and negatively charged residues, or no residues, the net phase accumulation would be negated. Thus to be able to perform phase unwrapping consistently it is necessary to set up some method of forcing paths to always encircle pairs of residues or no residues. A simple method that is able to achieve this is called the Nearest Neighbour Algorithm (NNA) (Cusack et al. 1995). Simply put, the algorithm attempts to find pairs of 'plus' and 'minus' residues and sets up barriers that the unwrapping algorithm cannot cross. These barriers are known as branch cuts. The method for this is both simple and effective:

1. Generate an array of every residue's charge and location by performing Equation (4.9) for each pixel in the phase image individually. All residues are then designated as unbalanced. See Figure 4.5a.

2. An unbalanced residue is then chosen from the array. When no such residue can be found, the algorithm is complete. See Figure 4.5e.

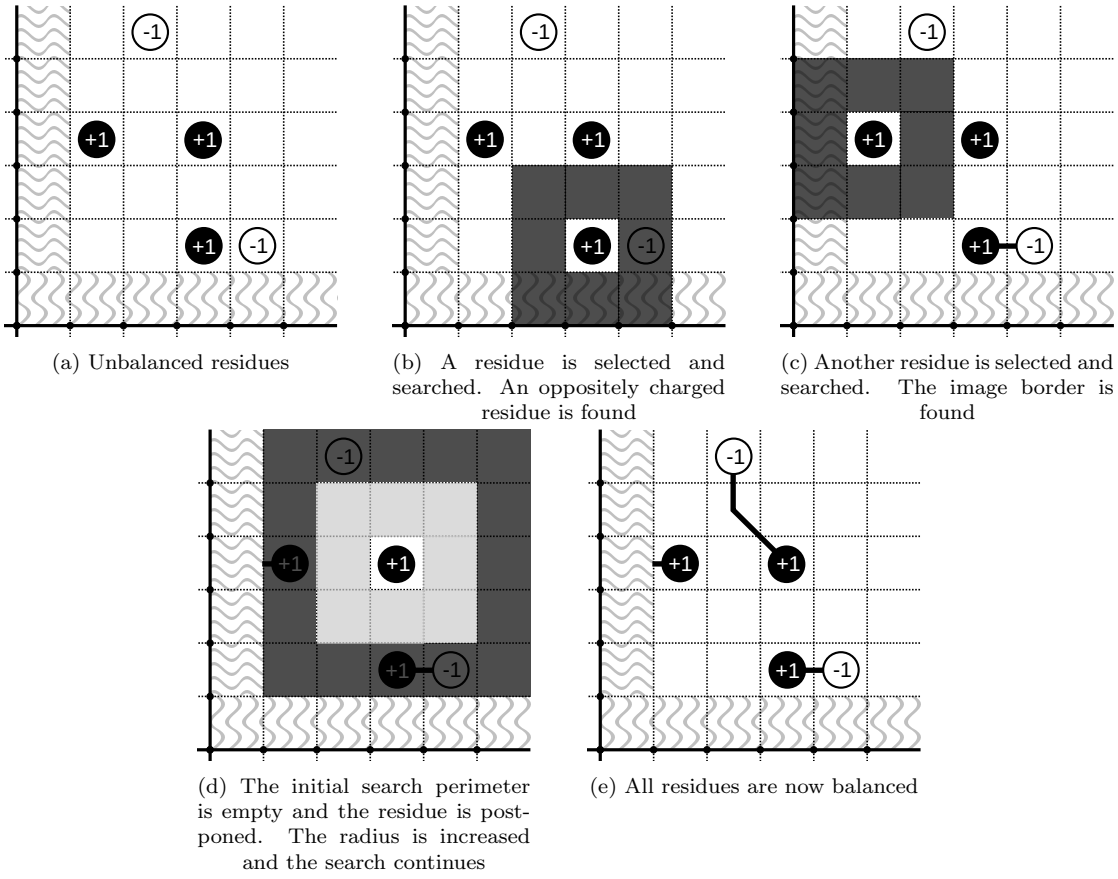
3. The chosen residue's surrounding pixel perimeter, initially its eight nearest neighbours, are searched for an oppositely charged residue that has not already been balanced. See Figure 4.5b.

   (a) If one such residue is found, a branch cut is drawn between them, that is the residues and the pixels between the residue are designated as part of a branch cut. Both residues are designated as balanced so they will be ignored in future searches. Repeat from step 2. See Figure 4.5b.

   (b) If the boundary of the image is found, the residue is connected to the image boundary and the residue is marked as balanced. Repeat from step 2. See Figure 4.5c.

4. If no candidate residues are found in the perimeter, the current residue is postponed and another unbalanced residue is chosen. Repeat from step 3. See Figure 4.5d.

5. If all remaining unbalanced residues have been postponed, increase the radius of the perimeter, e.g. from eight surrounding pixels to the 16 pixels that surround them. Repeat from step 2 until the perimeter radius has reached some predetermined size, usually such that it touches the image border. See Figure 4.5d.



(a) Unbalanced residues

(b) A residue is selected and searched. An oppositely charged residue is found

(c) Another residue is selected and searched. The image border is found

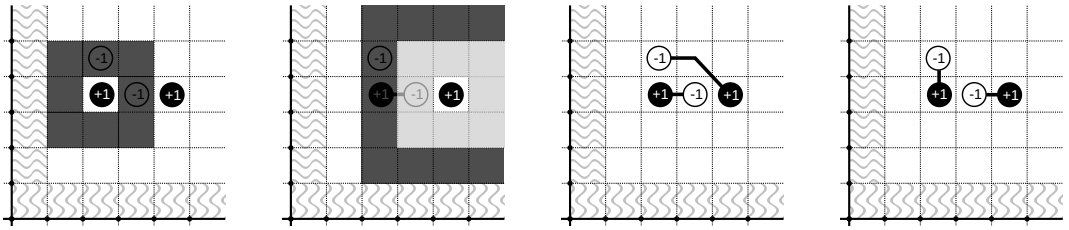
(d) The initial search perimeter is empty and the residue is postponed. The radius is increased and the search continues

(e) All residues are now balanced

Figure 4.5: Example of how the NNA sets branch cuts. The wavy lines represent the image border.

Figure 4.5 shows an example of how the NNA sets branch cuts between residues. The border of an image acts like an oppositely charged residue for any residue connected to the boundary, balancing the charge over all. The reason residues can be connected directly to the image border is that it prohibits the unwrapping algorithm from encircling the unpaired residue as it will not be able to pass between the residue and the border. The NNA is considered a greedy approach as each residue tries to minimise its own branch cut length but it does not necessarily minimise the cumulative branch cut length of all the residues. This leads to situations where residues are connected with unnecessarily long branch cuts, as shown in Figure 4.6. The problem arises when deciding how to choose where to draw branch cuts. For instance, in Figure 4.6a, there are two possible residues that are within the search perimeter and deciding which to connect to is not immediately obvious. The first solution, in Figure 4.6c, although correct, has a total branch cut length of six whereas the solution in Figure 4.6d has a length of four and as such is an optimal solution. Another choice that is not always obvious occurs when there is a possible residue and the image border in the search perimeter, as seen in

Figure 4.5b. In this situation the choice will result in the same total branch cut length but situations can arise where the choice is important.



(a) Selected residue has two possible connections

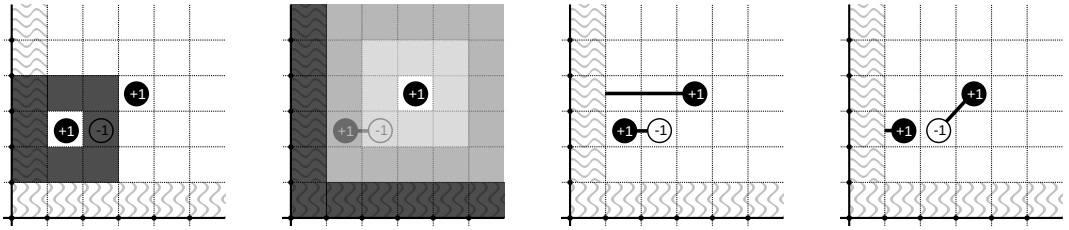
(b) The next residue is left with only one choice

(c) A possible, suboptimal solution

(d) Another solution with shorter total branch cut length

Figure 4.6: Example of how the NNA can create branch cuts of different lengths. The wavy lines represent the image border.

In Figure 4.7 the choice between the image boundary and an oppositely charged residue leads to a suboptimal solution, in Figure 4.7c, and an optimal solution, in Figure 4.7d. In this case connecting to the boundary first leads to the desired solution. The algorithm will always be biased towards one solution or the other, either you connect to the boundary first or the residue first. It is possible to overcome this problem by building every possible branch cut map associated with the image and selecting the map with the lowest total branch cut length. This of course will vastly increase processing time, especially for images with a high residue density.



(a) Selected residue can connect to the boundary or a residue

(b) The next residue is left with only one choice

(c) A possible, suboptimal solution

(d) Another solution with shorter total branch cut length

Figure 4.7: Example of how the NNA can create branch cuts of different lengths. The choice between the boundary and the possible residue is initially not obvious. The wavy lines represent the image border.

4.2.3 Modified Nearest Neighbour Algorithm

In Cusack et al. (1995) several other algorithms for placing branch cuts are investigated that attempt to improve the reliability of the solution. The stable marriage algorithm and simulated annealing both provided better solutions than the NNA but are hindered

by high noise density and prohibitively long running times, respectively. A modification to the NNA, called the Modified Nearest Neighbour Algorithm (MNNA), largely manages to overcome the problems of the other methods. The MNNA was found to be poorly optimised and would routinely create clusters of three residues, instead of pairs of positive and negative charges. The reason for this is due to the wording used in the method supplied in Cusack et al. (1995) that, in certain steps would lead to a runaway effect of joining residues together. Using a separate list of residues that are used in a particular step, that is updated at the beginning of that step only, this effect was curtailed. The modified methodology for the MNNA is as follows:

1. Generate an array of every residue's charge and location by performing Equation (4.9) for each pixel in the phase image individually.

2. Every residue is connected to either the nearest oppositely charged residue, regardless of whether the neighbouring residue already has a previous connection, or the image border if it is closer.

3. Find all residues that have more than one connection and add them to a list. For each residue in this list, if its connections each have only one connection of their own, find the next nearest neighbours, of opposite charge, of these connections. Connect the closest neighbours to their corresponding residues.

4. For every residue that is connected to both of the following:

   - residues which have more than one connection of their own, and
   - residues which have only the initial connection,

   the connections to the residues with multiple connections of their own are broken.

5. For every residue that is connected to both of the following:

   - the image boundary, and
   - another residue,

   sever the connection to the boundary. Repeat steps 4 and 5 until there are no more changes to the connections.

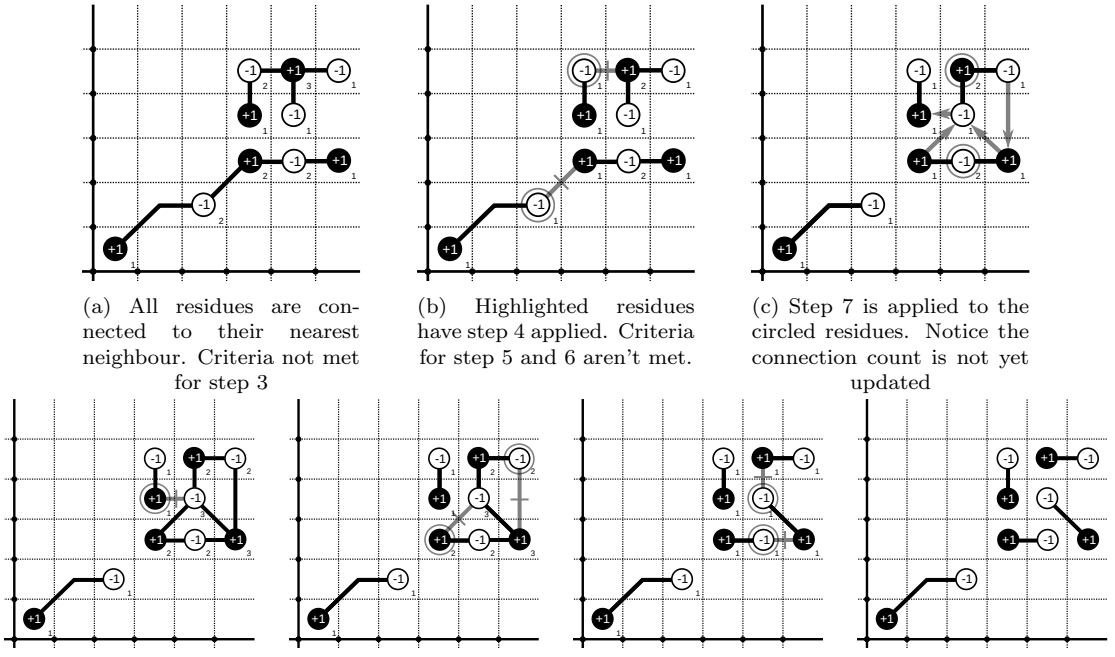
6. Find all residues that have exactly two connections and add them to a list. For every residue in the list, if its connections have two or more connections of their

own, break the longest connection of the current residue. If both connections are equally long then do not break either connection. Repeat steps 4 and 5 until there are no changes to the connections.

7. Find all residues that have more than one connection and add them to a list. For every residue in the list that has multiple connections, all of its partners, except the closest connection, are connected to their next nearest neighbour, of opposite charge, or the image boundary. If all the connections are equidistant then this is performed for all partners. Repeat steps 4 and 5 until there are no changes to the connections.

8. Repeat step 6.

9. There should now only be pairs of oppositely charged residues or residues connected to the boundary. The branch cuts can now be set.



(a) All residues are connected to their nearest neighbour. Criteria not met for step 3

(b) Highlighted residues have step 4 applied. Criteria for step 5 and 6 aren't met.

(c) Step 7 is applied to the circled residues. Notice the connection count is not yet updated

(d) Step 4 is applied to the circled residue. Criteria for step 5 aren't met.

(e) Step 6 is applied to the circled residues. Notice the connection count is not yet updated

(f) Step 4 is applied, for a final time, to the circled residues. Step 5's criteria aren't met

(g) All residues are now balanced

Figure 4.8: An example of how the Modified Nearest Neighbour Algorithm sets branch cuts

The modified nearest neighbour algorithm is a more robust method of joining residues together, however, it is still possible for it to fail in the presence of areas with high noise densities. It is reported by Cusack et al. (1995) that the method begins to fail at noise densities of 1 residue per 10 pixels.

4.2.4 Goldstein Branch Cut Method

A popular method that is able to overcome the various problems associated with the NNA is called the Goldstein-Zebker-Werner algorithm, or simply the Goldstein branch cut method (Goldstein et al. 1988). This method is able to produce short branch cuts, and as such optimal solutions, while retaining a fast processing time (Ghiglia & Pritt 1998). The biggest difference between the Goldstein method and the previous methods is that the NNA and MNNA always attempt to join two oppositely charged residues together, in a dipole configuration. The Goldstein branch cut method is able to create clusters of residues, with multiple positive and negative residues, with the stipulation that the total charge of the branch cut must be balanced. The Golstein branch cut method is quite similar to the NNA but with several important changes:

1. Generate an array of every residue's charge and location by performing Equation (4.9) for each pixel in the phase image individually. All residues are then designated as unbalanced and inactive. See Figure 4.9a.

2. An unbalanced residue is then chosen from the array. When no more unbalanced residues can be found, the algorithm is complete. See Figure 4.9e.

   (a) The residue is marked as active.

   (b) A counter is used to keep track of the cumulative charge of the branch cut. Initially it is set to the charge of the selected residue.

3. For each pixel in the chosen residue's surrounding perimeter, initially its eight nearest neighbours, search for an inactive residue or the image border. See Figure 4.9b.

   (a) If the boundary of the image is found, the residue is connected to the image boundary and the counter is set to zero, i.e. balanced. See Figure 4.9c.

   (b) If an inactive, unbalanced residue is found, the charge of the residue is added to the counter. The residue is then marked as active and balanced. They are then connected with a branch cut. See Figure 4.9b.

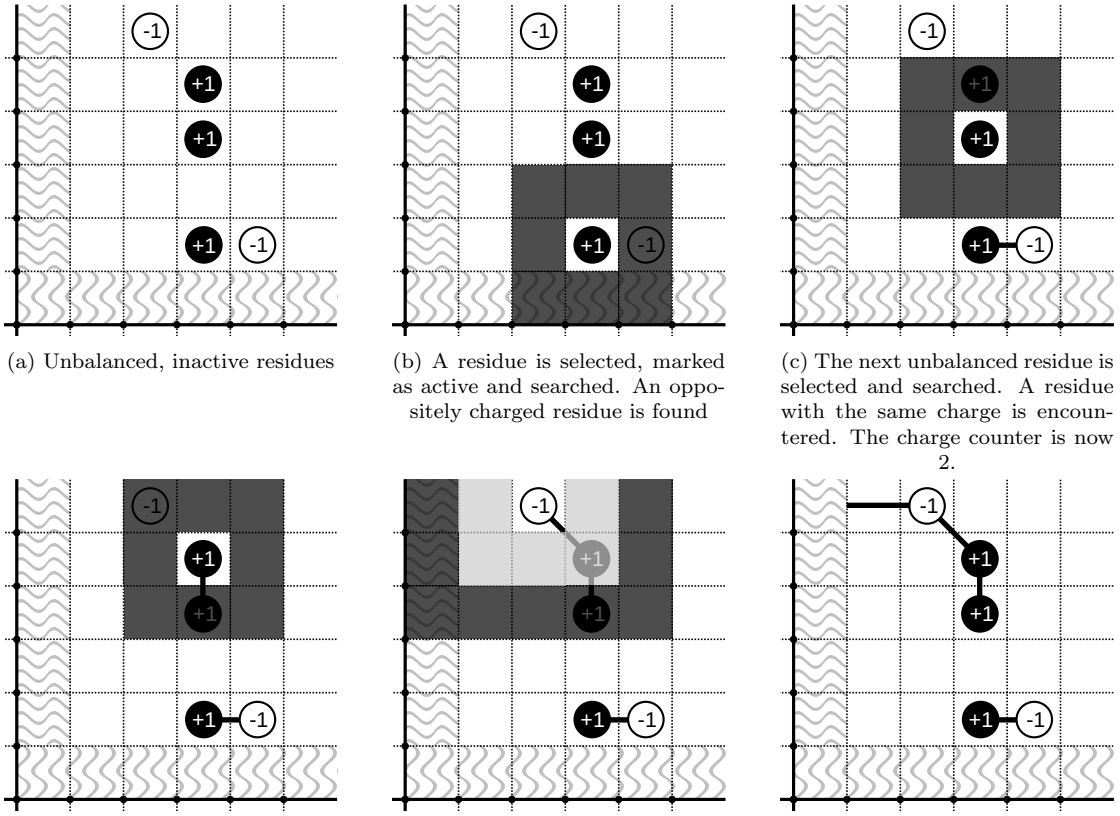
   (c) If an inactive, balanced residue is found, the residue is then marked as active. They are then connected with a branch cut. It is important to note that in this case the charge of the residues is not added as it is already balanced.

(d) Stop searching the perimeter and jump to step 6 if the counter has been balanced.

4. If there are no more pixels in the search perimeter, choose the next active residue and repeat from step 3.

5. If there are no more active pixels, increase the search radius and repeat from step 3 using the first active residue. Keep increasing the search radius as necessary until it reaches some predetermined size, usually such that it touches the image border.

6. All active pixels are marked as balanced and inactive. Repeat from step 2.



(a) Unbalanced, inactive residues

(b) A residue is selected, marked as active and searched. An oppositely charged residue is found

(c) The next unbalanced residue is selected and searched. A residue with the same charge is encountered. The charge counter is now 2.

(d) The next active residue is searched. An oppositely charged residue is found. The charge counter is now 1.

(e) After expanding the search perimeter, the image boundary is found. The counter is immediately set to zero.

(f) All residues are now balanced

Figure 4.9: Example of how Goldstein's branch cut method sets branch cuts. The wavy lines represent the image border.

Figure 4.10 shows how the NNA would perform if given the same residues. In Figure 4.9f it can be seen that the cumulative branch cut length is seven whereas in the NNA solution it is eight. The Goldstein branch cut method does not have the same problem as the

NNA of placing unnecessarily long branch cuts. This is because Goldstein's method allows an unbalanced residue to be attached to a residue that has already been balanced in the hopes that those residues will offer a more direct route to a boundary or to an oppositely charged residue to balance out the charge.

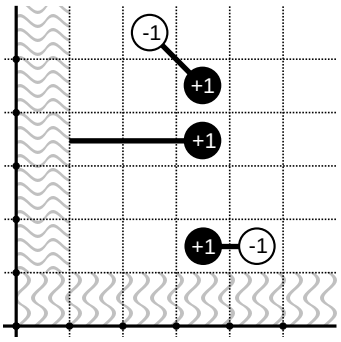


Figure 4.10: The solution from the Nearest Neighbour Algorithm when applied to the same residues as above.

4.2.5 Flood Fill Unwrapping

Having set the branch cuts it is now possible to unwrap the image. The most common algorithm used to perform this is known as a flood fill algorithm due to the way the algorithm "spreads out" over the image. A pixel, (i,j), is chosen in the image and acts as a reference point, for this reason it should be a pixel where the phase is known to be correct and is not on a branch cut. This can generally be achieved by choosing a pixel in an area with a high SNR. The initial pixel is marked as unwrapped and is set as the reference pixel. Its 4-connected neighbours, (i$\pm$1,j) and (i,j$\pm$1), are added to a queue of adjoining pixels. If any of these pixels are found to be on a branch cut they are removed from the queue. Each pixel from the queue is then investigated. It is unwrapped relative to a previously unwrapped neighbour, initially this will be the reference pixel only. Its nearest neighbours are then added to the queue, provided they are not on a branch cut, have already been unwrapped before, or are already in the list. The pixel is then dequeued and the next pixel is investigated. This is repeated until the queue is empty. Once all the valid pixels have been unwrapped it is then possible to unwrap the pixels that lie on branch cuts as the error will not propagate to any valid pixels.

In areas with a high noise density it is possible that several groups of residues may be clustered together which can lead to a situation where the flood fill algorithm cannot

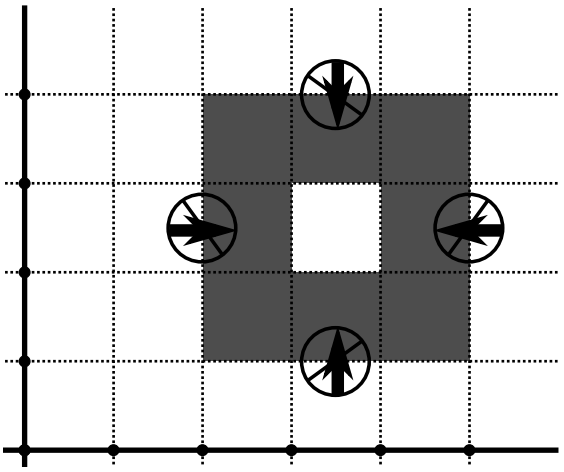


Figure 4.11: Branch cuts surrounding a pixel that cannot be reached during the unwrapping stage

reach part of the image as shown in Figure 4.11. In this figure the pixel in the centre of the branch cuts cannot be reached.

4.3 3D Extension of Branch Cut Method

The Goldstein branch cut method as well as the MNNA have trouble successfully setting up branch cuts in the presence of dense noise. It is possible to extend these methods into three, and higher, dimensions provided the data available supports it, i.e. the data is volumetric or the 2D slice is time-resolved. Doing so allows one to overcome the problem of unwrapping noisy areas. The methods described below are adapted from Huntley (2001) and Marklund et al. (2005).

As previously stated, in two dimensions a localised discontinuity source will create a pair of oppositely charged residues which can be avoided by setting up a cut between the two residues to force the unwrapping algorithm to encircle them both. In three dimensions, however, a localised discontinuity source will create a closed loop of residues surrounding the source unless the discontinuity is located near to the image boundary (Huntley 2001). The loop of residues is commonly known as a Phase Singularity Loop (PSL). In the 2D examples residues could only exist in one plane, perpendicular to the z-axis, however, in three dimensions there are now three planes, perpendicular to each major axis, on which to find residues. Equation (4.9) is therefore used to find residues perpendicular to the z-axis. By following the simple set of rules established by Equation (4.9) the methods for finding residues on the other, orthogonal planes may be derived. The rule

that is used to determine these equations is that for a right-handed Cartesian coordinate system. The pixels are investigated in a clockwise direction around the normal of the plane.

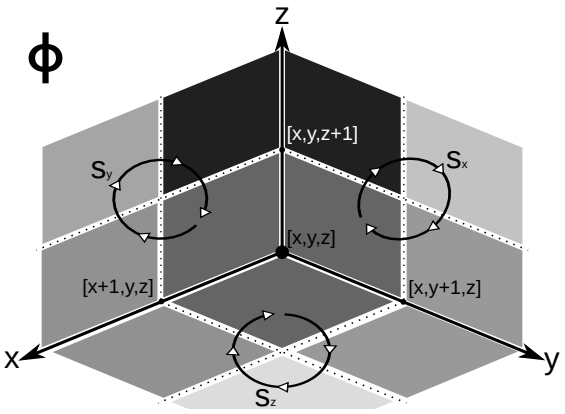


Figure 4.12: Cartesian set of axes showing the direction in which a net phase difference integral is applied for each major plane.

The closed loops marked s_z, s_y and s_x in Figure 4.12 represent the direction of the net phase difference integral for each respective major axis. The equations describing each integral are given in Equations (4.11) to (4.13).

$$s_z(x,y,z) = \lfloor \frac{\phi(x,y,z) - \phi(x,y+1,z)}{2\pi} \rceil + \lfloor \frac{\phi(x,y+1,z) - \phi(x+1,y+1,z)}{2\pi} \rceil + \dots \\ \lfloor \frac{\phi(x+1,y+1,z) - \phi(x+1,y,z)}{2\pi} \rceil + \lfloor \frac{\phi(x+1,y,z) - \phi(x,y,z)}{2\pi} \rceil \quad (4.11)$$

$$s_y(x,y,z) = \lfloor \frac{\phi(x,y,z) - \phi(x+1,y,z)}{2\pi} \rceil + \lfloor \frac{\phi(x+1,y,z) - \phi(x+1,y,z+1)}{2\pi} \rceil + \dots \\ \lfloor \frac{\phi(x+1,y,z+1) - \phi(x,y,z+1)}{2\pi} \rceil + \lfloor \frac{\phi(x,y,z+1) - \phi(x,y,z)}{2\pi} \rceil \quad (4.12)$$

$$s_x(x,y,z) = \lfloor \frac{\phi(x,y,z) - \phi(x,y,z+1)}{2\pi} \rceil + \lfloor \frac{\phi(x,y,z+1) - \phi(x,y+1,z+1)}{2\pi} \rceil + \dots \\ \lfloor \frac{\phi(x,y+1,z+1) - \phi(x,y+1,z)}{2\pi} \rceil + \lfloor \frac{\phi(x,y+1,z) - \phi(x,y,z)}{2\pi} \rceil \quad (4.13)$$

Example 4.1. *Figure 4.13 shows a* $3 \times 3 \times 3$ *excerpt from a larger 3D array of phase values; the values are presented as fractions of* π*.*

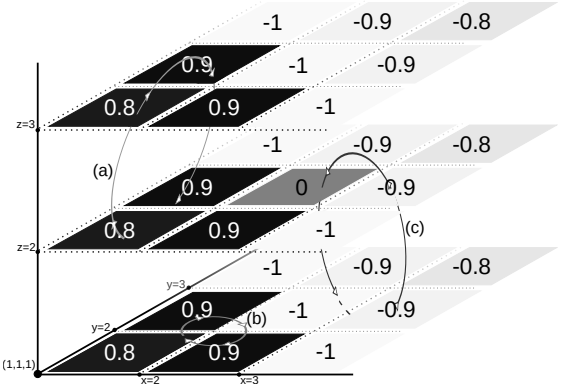


Figure 4.13: $3 \times 3 \times 3$ array of phase values. The three marked loops show (a) searching for an x-type residue at (1,1,2), (b) searching for a z-type residue at (1,1,1), and (c) searching for a y-type residue at (2,2,1).

The value at (2,2,2) creates a discontinuity source. This can be shown by applying Equations (4.11) *to* (4.13) *to the pixel and its neighbours. Examples of how these equations are applied are marked in Figure 4.13 from (a) to (c).*

$$s_z(1,2,2) = 1 \qquad s_y(2,2,1) = 1 \qquad s_x(2,2,1) = -1$$

$$s_z(2,1,2) = -1 \qquad s_y(2,2,2) = -1 \qquad s_x(2,2,2) = 1$$

After applying these equations to all of the pixels it is clear that the discontinuity at (2,2,2) has created six residues in the vicinity. In Example 4.3 we will be able to show that these residues form a closed PSL around the discontinuity.

The results from Example 4.1 highlight one of the problems with the 2D branch cut methods. These approaches, being limited to a single plane, are forced to almost randomly join residues together, often based on proximity, even though the residues may actually be unrelated. The inclusion of a third dimension makes it clear which residues are linked to one another as there will be an unbroken chain of residues forming a closed PSL which surrounds the discontinuity, only being broken if the discontinuity is near the image boundary, then known as a partial PSL. Each residue now has an extra characteristic besides its charge and location. That characteristic being its direction, or type, which points along the normal of the plane it was found on. A positive charge means the direction is parallel to the normal, and a negative charge means it is antiparallel to the normal. An $n \times 5$ list is used to store the n residues and their characteristics. The first column stores the residue type: 1 for s_x type residues, 2 for s_y type, and 3 for s_z type; the second column stores their charge: 1 or -1; and columns 3 to 5 are used to store the x-, y-, and z-coordinate, respectively. The residues actually exist in the centre of the face of the cube, as can be seen in Figure 4.14, and so the actual coordinates of s_1 would be (x+$\frac{1}{2}$,y+$\frac{1}{2}$,z). However, it does not make sense to store these $\frac{1}{2}$ values as these residues exist in an image with discrete pixel coordinate values. Therefore the standard that has been adopted is to use the bottom left corner of the face's coordinate values; so the coordinates of s_1 will be stored as (x,y,z).

Before the residues can be connected together it is important to define the following five arrays. First, a mask array must be defined. The mask array must be the same size as the phase volume and store a 1 for each pixel which is in the mask and a 0 for pixels which aren't. An array of the same size is then created based on the mask, called the phase boundary. The phase boundary is used when closing partial PSLs, discussed in Section 4.3.2.

The phase boundary is generated based on the following rules: for a pixel (x,y,z) in the mask array with the value 1, if any of its 26 nearest neighbours, that is (x±1,y±1,z±1), has a value of 0 in the mask array, then the pixel (x,y,z) is set to 1 in the phase boundary array. Otherwise the pixel has a value of 0.

The last three arrays that must be defined are flag arrays for each major axis, namely F_x, F_y, and F_z. These flag arrays are used to decide where the unwrapping algorithm can and cannot go. F_x, F_y, and F_z, will be used to block movement between adjacent pixels in their respective subscripts' directions. For example, if $F_x(x, y, z)$ is set to 1 it

will block any direct movement between (x,y,z) and (x+1,y,z). For each pixel (x,y,z) in the mask, the flag arrays are generated as follows:

$$F_x(x,y,z) = mask(x,y,z) \oplus mask(x+1,y,z)$$

$$F_y(x,y,z) = mask(x,y,z) \oplus mask(x,y+1,z)$$

$$F_z(x,y,z) = mask(x,y,z) \oplus mask(x,y,z+1)$$

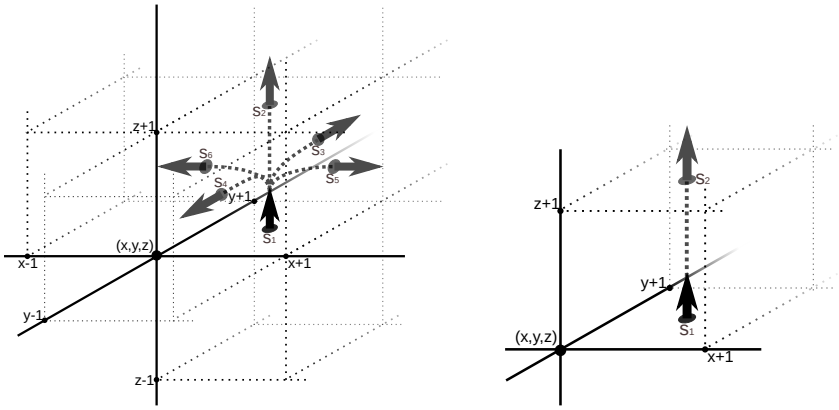
Where $\oplus$ is the exclusive or, XOR, operator.

4.3.1 Connecting Residues

Example 4.2 can be applied to all types and charges of residues. Lookup tables of all the possible connections that can be made from a given residue to another are found in Appendix A. Table A.1 gives the possible connections that follow the direction of the type and charge of the residues and are called forward connections. Connections can also be made in a backward direction, as a forward connection from the other residue's perspective is a backward connection. In Table A.2 the lookup table for backward connections is given. Backward connections are used for cases where PSLs terminate on the phase boundary: once all the forward connections have been made from an initial residue, the loop is found to terminate on the phase boundary. The backward connections of the initial residue are then investigated until it too terminates on the phase boundary.

Example 4.2. *The residue entering the cube at (x,y,z) from below is labelled* s_1 *in Figure 4.14a:*

Label	Type	Charge	x-	y-	z-coordinate
s_1	3	1	x	y	z



(a) Possible connecting residues from a positive, z-type residue.

(b) A simple case for joining two residues.

Figure 4.14: Connecting a residue to its candidate residues

For there to be an unbroken loop surrounding the discontinuity, the loop must exit through at least one of the five remaining faces of the cube, labelled s_2 *to* s_6*.*
This means that s_1 *can only be joined to the following five residues:*

Label	Type	Charge	x-	y-	z-coordinate
s_2	3	1	x	y	z+1
s_3	2	1	x	y+1	z
s_4	2	-1	x	y	z
s_5	1	1	x+1	y	z
s_6	1	-1	x	y	z

If, of the possible residues above, only s_2 *was valid then it would be joined to* s_1 *unambiguously; this is shown in Figure 4.14b.*

Not all of the five possible connections for a given residue will necessarily exist, however, at least one connection will correspond to a real residue. If none of the possible connections correspond to real residues, then the residue under investigation must connect with the phase boundary.

4.3.2 Sorting Residues

The list of residues is initially unsorted and as such it is not known which residues belong to which loop. Therefore it is useful to keep track of the order of the residues in a PSL as well as the type of PSL, whether it is closed or partial. An ordered list can be created during the connection stage. As a residue is put under investigation, it is added to a new list with the same structure as the unsorted list, an $n \times 5$ matrix. When a new connecting residue, found using Table A.1, is found it is placed below the previous residue. This is continued until a residue is found that connects back to the initial residue or ends on the phase boundary. If the boundary is found, return to the initial residue that was investigated and apply Table A.2 to find residues that it can connect to backwards. Each time such a residue is found it is placed above the preceding residue, this is repeated until a residue is found that terminates on the boundary.
It is also important to keep track of the number of distinct PSLs as well as which residues actually make up a given PSL. An $m \times 3$ pointer list can be used for this purpose, where m is the number of distinct PSLs. For each row, or PSL, the first column is used to store the index, from the sorted list, of that PSL's first residue; the second column stores the index of the last residue in the PSL; and the third column stores what type of PSL it is, 0 for a partial PSL and 1 for closed.

Example 4.3. *Continuing Example 4.1 we will now be able to show that the resulting six residues form a closed loop PSL. The table below represents the $n \times 5$ list that is used to store the unsorted residues. Figure 4.15 shows a graphical representation of the residues as they might appear on 3D Cartesian axes.*

Unsorted list of residues

Label	Type	Charge	x	y	z
s_1	3	1	1	2	2
s_2	3	-1	2	1	2
s_3	2	1	2	2	1
s_4	2	-1	2	2	2
s_5	1	-1	2	2	1
s_6	1	1	2	2	2

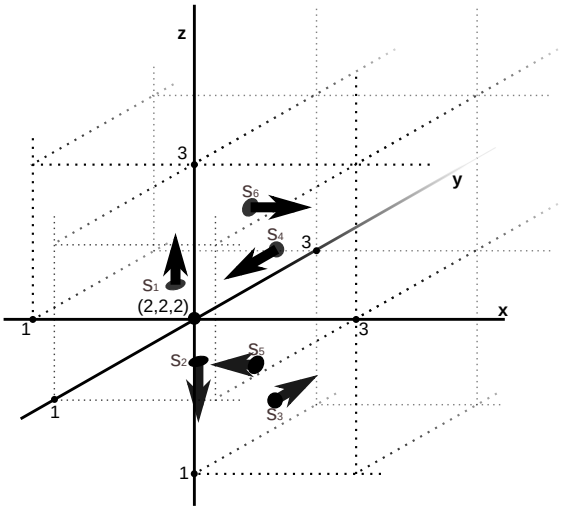


Figure 4.15: 3D Cartesian axes showing residues from Example 4.1. The residues are labelled from s_1 to s_6.

Starting with the first residue, s_1, in the unsorted list above, we apply the lookup Table A.1. The five possible connecting residues are as follows:

Possible forward connections from residue s_1

	Type	Charge	x	y	z	Label
1	1	1	2	2	2	s_6
2	1	-1	1	2	2	N/A
3	2	1	1	3	2	N/A
4	2	-1	1	2	2	N/A
5	3	1	1	2	3	N/A

The first candidate residue is entry s_6 in the unsorted list of residues and so can be connected to the original residue, s_1. Residue s_6 now becomes our current residue and

Table A.1 is applied to it. This method is repeated until we return to the original residue, s_1. In this case the order in which the residues are connected is:

Table 4.1: Sorted list of residues

Label	Type	Charge	x	y	z
s_1	3	1	1	2	2
s_6	1	1	2	2	2
s_4	2	-1	2	2	2
s_2	3	-1	2	1	2
s_3	2	1	2	2	1
s_5	1	-1	2	2	1

Table 4.2: Pointer list for sorted residues

Loop ID	1^{st} index	Last index	Loop type
1	1	6	1

In Figure 4.16 the residues are shown as they would appear when they are connected together.

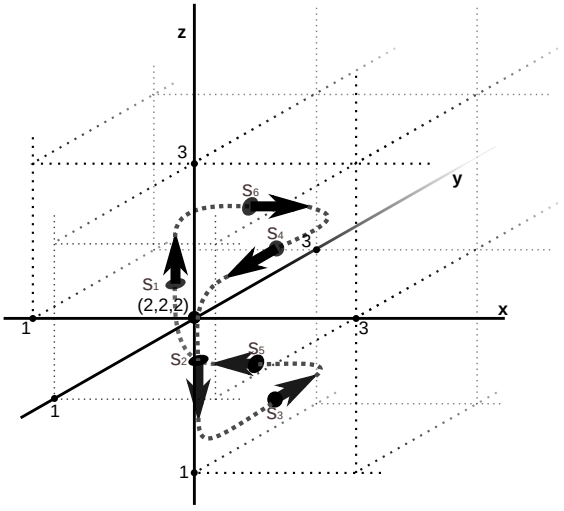


Figure 4.16: 3D Cartesian axes showing residues from Example 4.1 after they have been connected together. The residues are labelled from s_1 to s_6.

Examples 4.1 and 4.3 give a very simplified case where all the residues found were directly related to each other. In general an image can contain hundreds of discontinuity sources and as such will have several times that number of residues, the vast majority of which will be unrelated to each other.

Knot points

So far only the simplest, and most common, case for residues have been discussed, that is one residue "entering" a cube and one "exiting", creating an unambiguous connection. It is, however, possible for a given cube to have several residues entering and exiting. These residues are then said to form a knot where the cube is the knot point. Because a residue must be able to connect to only one other residue, there are always an even number of residues associated with a cube and an equal number of entering and exiting residues. Each face of the cube can have at most one residue which means a cube can have a maximum of six residues, three entering and three exiting. If there is an imbalance in the number of entering or exiting residues at least one of the faces of the cube must be touching the phase boundary.

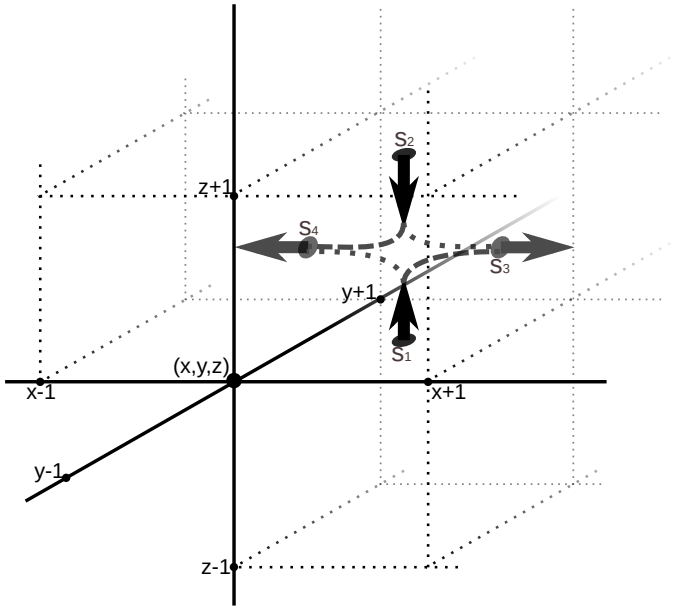


Figure 4.17: A knot point. The dashed and dotted lines show the two possible ways of connecting the residues.

In Figure 4.17 there are two residues, s_1 and s_2, entering the cube. The two residue s_3 and s_4 are exiting the cube. It is now no longer clear which residues should be connected together as both s_1 and s_2 can be connected to either s_3 or s_4. Of course only one residue

may be connected to each so when a connection is made there can be only one other viable option for the remaining residues as is indicated by the dashed and dotted lines.

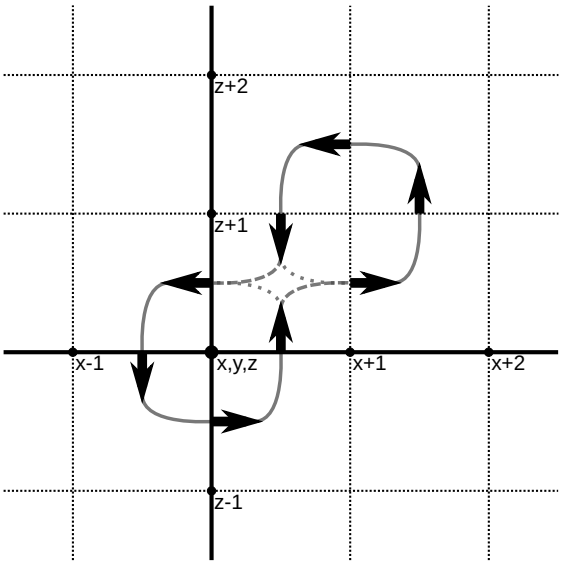


Figure 4.18: A 2D view down the y-axis of a knot point and the possible resulting PSLs.

When a knot is encountered there may be multiple loops that a residue can be a part of. In a physical sense it is important how a knot is treated during the sorting stage, for instance in Figure 4.18 there are three possible PSLs that will result from the choice of connections. If the dotted connections are chosen there will be two distinct closed PSLs, however, if the dashed connections are made there will be only one closed PSL that encompasses all the residues in the image. One method for dealing with knots is to build all possible PSLs and select the shortest loop (Marklund et al. 2005). This can become time consuming especially for areas with a high noise density. Although physically there is a difference between the two sets of PSLs, there is no difference between the two choices when the branch cuts are being set, discussed in Section 4.3.3 (Huntley 2001). This means that connections can actually be made arbitrarily. Although, if during the investigation of the knot point a connection is found to be part of a partial PSL, one, or more, of the other residues in the knot point must also be part of a partial PSL and they must, by necessity, be joined together. This must be done so that partial PSLs aren't accidentally attached to closed PSLs.

Closing Partial Phase Singularity Loops

It is necessary to close partial PSLs as the branch cut building algorithm, discussed in Section 4.3.3, needs to be applied to closed PSLs. The method for this involves creating artificial residues to join the ends of partial loops together. The artificial residues must be chosen to minimise the distance between the two ends of the partial PSL while only going through the phase boundary.

A partial PSL is selected from the pointer list and the last residue in the loop is investigated using Table A.1. This residue terminates on the phase boundary so it will not be possible to make any connections to real residues. Instead the five possible connections will be designated as possible artificial residues. Not all of these artificial residues will be viable candidates, only artificial residues whose coordinates are in the phase boundary are considered viable. One artificial residue is selected from the list of viable candidates such that it minimises the distance between the current residue and the first residue in the partial PSL. That artificial residue is then added to the list, below the current residue, and it is then investigated using the same process as above. Finally, when an artificial residue is found that connects directly back to the first residue of the partial PSL under investigation it is added to the list. The pointer list is then updated to reflect that the partial PSL has now been closed and that there are new "residues" in the loop. The next partial PSL in the pointer list is then chosen and the process is repeated until all the partial PSLs have been closed.

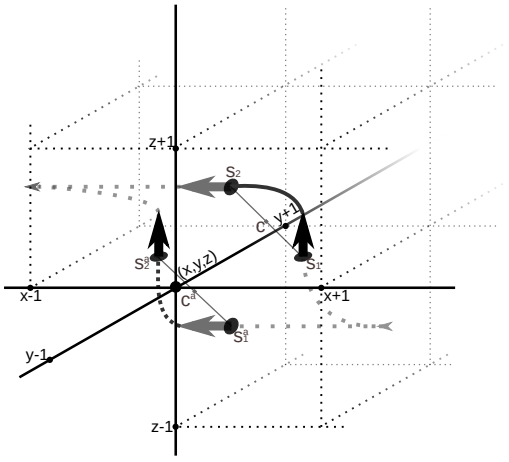
4.3.3 Branch Cuts

In the 2D example of phase unwrapping, branch cuts are lines that join oppositely charged residues to force an unwrapping algorithm to go around pairs of residues to stop phase errors from being propagated throughout an image. In 3D space there are now closed loops surrounding areas of noise and so, by analogy, an unwrapping algorithm should be stopped from directly passing through a PSL. Therefore, instead of lines, the branch cut building algorithm will set up branch cut surfaces that block paths through the PSLs. The previously defined flag arrays, F_x, F_y, and F_z, are now used to block movement of the unwrapping algorithm when it would pass through a PSL. The arrays are filled by shrinking the PSL towards its centre and as the boundary of the loop

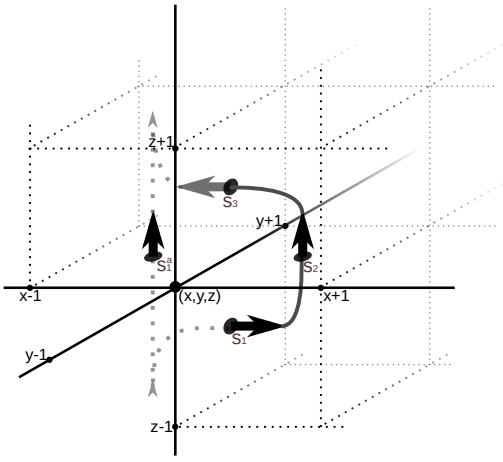
'cuts through' a vertex connecting two pixels, movement along that vertex is blocked by setting a flag in the appropriate flag array.

Reducing the Loop Size

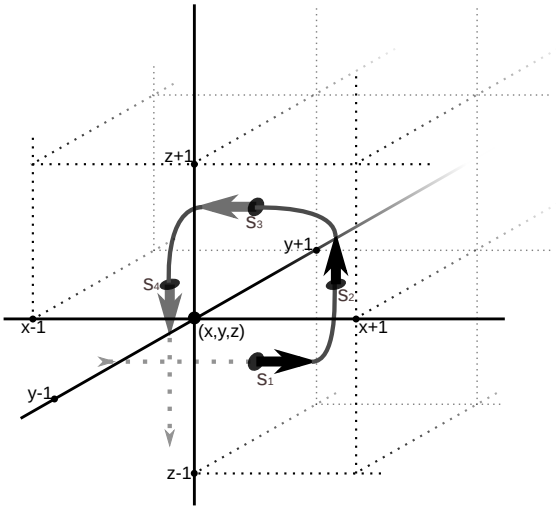
To shrink the loop towards its centre it is necessary to remove groups of residues and replace them with new artificial residues in such a way that it reduces the area of the branch cut surface without breaking the loop. This can be achieved by finding three particular types of turns that can be found in a loop. The three type of turns are illustrated in Figure 4.19.



(a) A 90° turn and possible replacement artificial residues. The centre of each turn is marked c and c^a, respectively.



(b) A 180° turn and possible replacement artificial residue.



(c) A 270° turn will create a knot point.

Figure 4.19: Three types of turns found in PSLs.

90° Turns Finding a 90° turn involves inspecting the directions of two connected residues. If their directions are not the same, that is not a straight line, it is a 90° turn. Figure 4.19a shows an example of a 90° turn: Residue s_1 points along the z-axis and s_2 points along the x-axis.

Replacing a 90° turn involves removing the residues s_1 and s_2, and replacing them with two new artificial residues s_1^a and s_2^a.

The rules for choosing the direction and charge of the new artificial residues are simple: s_1^a will have the same direction and charge as s_2, and s_2^a will have the same direction and charge as s_1.[1]

$$s_1^a(1:2) = s_2(1:2)$$
$$s_2^a(1:2) = s_1(1:2)$$

The coordinates of the new artificial residues must be chosen such that they maintain the closed nature of the loop which means the rules for determining these coordinates are: The coordinates of s_1^a and s_2^a are the same as the coordinates of s_2 and s_1, respectively, however, an offset is applied to each set of coordinates based on the direction and charge of s_1 and s_2, respectively. This process is shown below.

$$\text{offset} = \begin{matrix} 1\ 0\ 0 \\ 0\ 1\ 0 \\ 0\ 0\ 1 \end{matrix}$$

$$s_1^a(3:5) = s_2(3:5) - s_1(2) \times (\text{offset}(s_1(1), :))$$
$$s_2^a(3:5) = s_1(3:5) + s_2(2) \times (\text{offset}(s_2(1), :))$$

The actual replacement of a 90° turn, however, shall only be performed when the addition of the two new artificial residues causes a reduction in the area of the loop. The addition of the new residues should always shrink the loop towards its geometric centre. Determining whether this occurs can be achieved by finding the centre of the original 90° turn, marked c in Figure 4.19a, the centre of the new, artificial turn, marked c^a, and the geometric centre of the loop. The centre of each can be found by averaging the

[1] Please note these equations use MATLAB's (Mathworks, Natick, MA) array numbering conventions.

coordinates of the associated residues. If

$$\|c_{geom} - c^a\| < \|c_{geom} - c\|$$

then the artificial residues can be inserted to replace s_1 and s_2, where $\|\cdot\|$ is the Euclidean norm operator and c_{geom} is the geometric centre of the loop.

180° Turns A 180° turn is made up of three connected residues, as shown in Figure 4.19b. The directions of s_1 and s_3 must be the same while the direction of s_2 must not be the same, and the charge of s_1 must be the opposite of the charge of s_3 in order for the three connected residues to comprise a 180° turn.

$$\begin{aligned} s_1(1) =& s_3(1) \neq s_2(1) \\ s_1(2) \neq& s_3(2) \end{aligned}$$

Only one artificial residue, s_1^a, is used to replace the residues s_1 to s_3. The artificial residue s_1^a will have the same charge and direction as s_2. As with 90° turns, the coordinates of the new artificial residue must maintain the closed nature of the loop. The coordinates of s_1^a are the same as the coordinates of s_2 with an offset applied based on the direction and charge of s_3.

$$\begin{aligned} s_1^a(1:2) =& s_2(1:2) \\ s_1^a(3:5) =& s_2(3:5) + s_3(2) \times (\text{offset}(s_3(1), :)) \end{aligned}$$

The artificial residue must always reduce the area of the loop which implies that

$$\|c_{geom} - s_1^a(3:5)\| < \|c_{geom} - s_2(3:5)\|$$

If this criterion is met the artificial residue, s_1^a, can be used to replace residues s_1 to s_3.

270° Turns A 270° turn is made up of four connected residues and always forms a knot, as can be seen in Figure 4.19c. The rules for finding 270° turns are very similar to those for finding 180° turns. The directions of s_1 and s_3 must be the same, and s_2 and

s_4 must have the same direction but these pair of directions must not be the same. The charges of s_1 and s_3 must oppose as must the charges of s_2 and s_4.

$$s_1(1) = s_3(1) \neq s_2(1) = s_4(1)$$
$$s_1(2) \neq s_3(2)$$
$$s_2(2) \neq s_4(2)$$

In the case of a 270° turn, all of the residues that it comprises can be removed. Referring back to Figure 4.18 a 270° turn can be treated like a closed loop encircling a single point, as it could have been if a different connection was made. For instance if in Figure 4.18 residues s_4 and s_1 were directly connected to form a closed loop with no knot. It is for this reason that connections in knot points can be made arbitrarily.

Clearly a 90° turn is the most general type of turn, as each other turn is simply a string of concurrent 90° turns. For this reason during implementation it is important that turns be removed in decreasing order of specificity. The following list describes the order in which turns should be removed.

1. If only four residues surrounding a single vertex remain, the loop has been shrunk to its smallest size and the operation is complete. Otherwise proceed to step 2.
2. Search for 270° turns.
    - If such a turn is found remove it and return to step 1.
    - Otherwise if there are no more 270° turns in the loop proceed to the next step.
3. Search for 180° turns.
    - If found attempt to remove and replace the residues using the previously discussed method.
        - If successful, return to step 1.
        - If unsuccessful, mark the turn as a recently visited invalid 180° turn and continue searching.
    - If there are no more valid 180° turns continue to the next step.

4. Search for 90° turns.

   - If a 90° turn is found, that is not part of an invalid 180° turn, attempt to remove and replace the residues using the previously discussed method.
     - If successfully replaced, return to step 1.
     - If unsuccessful, mark it is as a recently visited invalid 90° turn and continue searching.
   - If there are no more valid 90° turns move onto the next step.

   At this point the loop comprises more than four residues and there are no more valid moves.

5. Find the most recently visited invalid 90° turn

   - If it exists, immediately replace it and return to step 1.
   - If it no longer exists due to a previous replacement begin a search for a 90° turn, that is not part of an invalid 180° turn, and replace it. Return to step 1.
   - If no such 90° turns exist continue to the next step.

6. Find the most recently visited invalid 180° turn

   - If it exists, immediately replace it; return to step 1.
   - If it no longer exists due to a previous replacement search for a new 180° turn and replace it; return to step 1.

Placing Branch Cuts

During the shrinking operation the flag arrays, F_x, F_y and F_z, are filled. From Figure 4.19 it can be seen, for any given turn, that all the residues and associated cube faces share one common vertex. In this case it is the vertex that connects points (x,y,z) to (x,y+1,z) for all the turns. During the shrinking operation a turn may be removed and replaced with new artificial residues. When this occurs the shrinking operation can be thought to "cut" through a vertex of the cube. Whenever such a "cut" is made a flag is set so that the unwrapping algorithm cannot directly move along that vertex. So for the case of all the turns in Figure 4.19, $F_y(x, y, z)$ would be set to 1 so that during the

unwrapping operation there will be no direct movement between (x,y,z) and (x,y+1,z). This common vertex can be found quite simply. When any valid turn is found and replaced, the first two residues s_1 and s_2 are investigated. s_1 and s_2 will always have different directions, the appropriate flag array is chosen by the direction that is common to neither. For instance, if the direction of s_1 is 1, x-direction, and s_2 is 3, z-direction, the flag array F_y is used. The maximum value of the x-, y-, and z-coordinates of s_1 and s_2 are used to determine the x-, y-, and z-coordinates, respectively, of the flag array which must be marked as 1.

$$\begin{aligned} s_1(1) \neq & d \neq s_2(1) \\ x_f = & \max(s_1(3), s_2(3)) \\ y_f = & \max(s_1(4), s_2(4)) \\ z_f = & \max(s_1(5), s_2(5)) \end{aligned}$$

and so,

$$f_d(x_f, y_f, z_f) = 1$$

This is performed each time a turn is removed and replaced. When the loop is shrunk to its smallest size, a loop containing exactly four connected residues, it will encircle one vertex which is removed using the same rules as above.

The PSL has been used to set its branch cut surface and is now complete and can be removed from the pointer list of PSLs. After all the PSLs have had their branch cut surfaces set, the unwrapping algorithm can be applied.

Example 4.4. *Continuing Example 4.3 we can now show how this particular branch cut surface is reduced. Below we have the sorted residues from Figure 4.16 and the pointer list. The first PSL is selected from the pointer list and the first residue is investigated using the method described previously.*

Table 4.3: Sorted list of residues

Label	Type	Charge	x	y	z
s_1	3	1	1	2	2
s_6	1	1	2	2	2
s_4	2	-1	2	2	2
s_2	3	-1	2	1	2
s_3	2	1	2	2	1
s_5	1	-1	2	2	1

Table 4.4: Pointer list for sorted residues

Loop ID	1^{st} index	Last index	Loop type
1	1	6	1

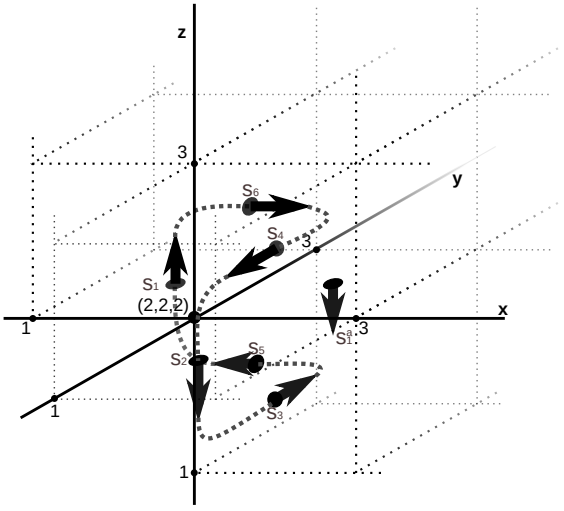


Figure 4.20: A 180° turn is found and the artificial residue, s_1^a, can replace the residues s_4, s_2, and s_3.

There are no 270° turns so 180° turns are investigated. The first 180° turn is found at s_4, and is comprised of residues s_4, s_2, and s_3 as shown in Figure 4.20. The new artificial residue, s_1^a, is found to reduce the size of the loop and so residues s_2 to s_4 are removed and s_1^a is connected to the remaining residues as shown in Figure 4.21.

The removal of the 180° turn means a flag needs to be set. The turn was comprised of two y-type residues and a z-type residue which means the flag is set in F_x, specifically at (2,2,2). This will block any direct movement between the pixels at (2,2,2) and (3,2,2).

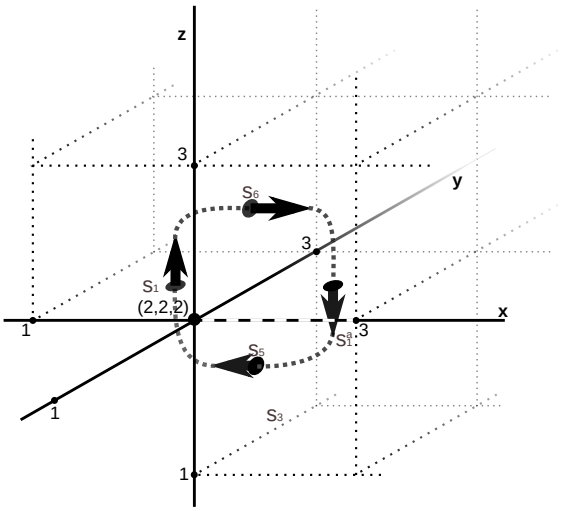


Figure 4.21: The 180° turn is removed and the artificial residue, s_1^a, is connected to residues s_5 and s_6. The vertex between (2,2,2) and (3,2,2) is flagged.

There are now exactly four residues remaining. This means that the PSL has been reduced to its smallest size and these residues can be removed and a flag is set. There are two z-type and two x-type residues so a flag must be set in the F_y array at (2,2,2) which can be seen in Figure 4.22.

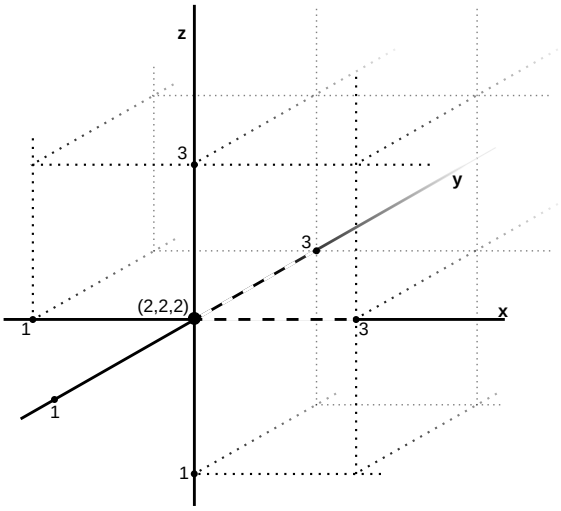


Figure 4.22: The selected PSL has been reduced to its smallest size and all necessary flags have been set.

4.3.4 Flood Fill Unwrapping

The flood fill unwrapping algorithm is very similar to the algorithm described in Section 4.2.5, the major difference being that it must now operate in three spatial dimensions. A voxel, (i,j,k), is chosen as a reference point and the 6-connected neighbours, {(i±1,j,k);(i,j±1,k);(i,j,k±1)}, are added to a list of neighbours to unwrap. If unwrapping along a vertex is forbidden by one of the flag arrays then the neighbour on that vertex is removed from the list. For example, if $F_x(i,j,k) = 1$ then the neighbouring voxel (i+1,j,k) should be removed from the list, or if $F_x(i-1,j,k) = 1$ then the neighbouring voxel (i-1,j,k) should be removed. Once only valid neighbours are left in the list, the first neighbour is unwrapped with respect to a previously unwrapped neighbouring voxel, initially this will be the reference voxel. Its neighbours are then added to the list, again checking the flag arrays to determine if a neighbour is valid. If a neighbour is already in the list or has been unwrapped before do not add it to the list. The voxel is then marked as unwrapped and removed from the list of voxels to unwrap. This process is then repeated until the list of neighbours is empty and all voxels have been unwrapped.

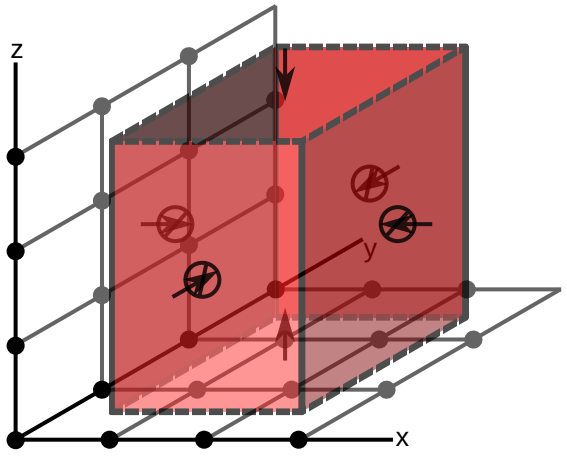


Figure 4.23: A group of branch cut surfaces that stop movement along the x-, and y-directions but allow it along the z-direction.

The Goldstein branch cut method and the modified nearest neighbour algorithm could leave parts of the image unable to be unwrapped when branch cuts surround noisy areas. The 3D extension and its branch cut surfaces may allow these areas to be unwrapped if there is a valid direction, that is it is not blocked by a branch cut surface. For instance, a branch cut surface could exist such that it stopped the flood fill algorithm from unwrapping along the x-, and y-directions, as in the case shown in Figure 4.23,

however, it would then still be able to unwrap the isolated voxels along the z-direction. Clearly if this were treated as individual 2D slices there would be several voxels within the border created by the group of branch cut surfaces that the flood fill algorithm would not be able to reach.

4.4 Quality Guided Phase Unwrapping

The previous sections have detailed spatial methods of unwrapping phase maps based on path following. They involve setting up branch cuts or areas where an unwrapping algorithm cannot go. In doing so phase discontinuities are avoided and as such phase errors are isolated to small, noisy regions. The method for these algorithms may appear simple in theory but the actual implementation of the algorithm can become very complex. Slow running times are also common for large datasets. The algorithms are also highly susceptible to noise and begin to fail more regularly as the SNR of the image drops.

A simpler approach to spatial unwrapping can be achieved by using the quality of the phase maps, an informal measure of the map's SNR, to guide the unwrapping algorithm through areas of high quality phase first and then lower quality areas; this is known as Quality Guided Phase Unwrapping (QGPU). Several authors, Cusack & Papadakis (2002), Ghiglia & Pritt (1998), Lu et al. (2012), Suprijanto et al. (2001), Zhang et al. (2014), and Zhou et al. (2009), have proposed methods for QGPU algorithms. The basic assumption that all QGPU algorithms use is that low quality areas, usually due to low SNR, contain the majority of phase discontinuities and so should be avoided initially. The algorithm is guided through high quality areas first to isolate these low SNR areas and therefore stop phase errors from being propagated throughout the image. They are also able to overcome many problems that affect the path following algorithms especially problems caused by areas with low SNR.

4.4.1 Pole Field Noise Estimation

Cusack & Papadakis (2002) used a method for estimating noise that uses many of the same algorithms used by branch cut path following methods. Residues, or poles as they are referred to in (Cusack et al. 1995), are found using Equations (4.11) to (4.13). If a

residue is found it is marked with a 1, as the direction of the charge is not important in this instance. It is placed in the centre of the face that it was found in, this is to avoid any confusion when different types of residues are found at the same location. For example an x- and z-type residue found at (x,y,z) would be placed at (x,y+$\frac{1}{2}$,z+$\frac{1}{2}$) and (x+$\frac{1}{2}$,y+$\frac{1}{2}$,z), respectively. All poles are placed in a single volume called the pole field. The pole field is then smoothed by performing a 3D convolution using a smoothing filter. Cusack & Papadakis (2002) used a pseudo-Gaussian smoothing kernel as it can be applied to each axis separately, avoiding the use of a 3D convolution.

The smoothed pole field is then used as a mask during the unwrapping stage. The unwrapping stage uses a modified version of the algorithm described in Section 4.3.4. It uses a threshold variable to force the unwrapping algorithm to only unwrap voxels that are less than or equal to the threshold value. The threshold varies between the minimum and maximum values taken from the smoothed pole field. Once all the voxels that have a value less than the threshold value have been unwrapped the threshold is increased and any new voxels are unwrapped with respect to previously unwrapped voxels. This is repeated until the threshold reaches its maximum value at which point the last voxels are unwrapped. The level of smoothing and the number of threshold steps can be varied parametrically to fine tune the algorithm.

This method has been shown to be robust by Cusack & Papadakis (2002), however, it is not a true measure of phase quality and is still susceptible to some problems that face branch cut methods, such as seed selection problems. Seed selection is generally chosen by finding the voxel with the highest quality value and unwrapping with reference to it. It is possible that such a voxel may be surrounded by an area of "low quality" voxels which would mean that the threshold value would have to be raised before the unwrapping algorithm would be able to unwrap neighbouring pixels. This can become a problem if there are areas of "higher quality" that are isolated from the seed voxel because they are only reachable after the threshold is raised, meaning phase errors from the "low quality" areas may be propagated into these "high quality" areas possibly corrupting the phase values.

4.4.2 Pseudocorrelation

Correlation maps are commonly used by Synthetic Aperture Radar images. They have access to several complex images taken from multiple reference points at a single time

point. Pseudocorrelation maps are based on these correlation maps except, like in the case of MRI, there is only access to one reference image at a time (Suprijanto et al. 2001). The pseudocorrelation maps show areas with strong correlations as high quality and weak correlation as low quality. The map is generated using Equation (4.14).

$$|q_{\mathrm{pc}}(x,y)| = \frac{\sqrt{\left(\sum_{\substack{x-k\leq i\leq x+k\\ y-k\leq j\leq y+k}} \cos\phi(i,j)\right)^2 + \left(\sum_{\substack{x-k\leq i\leq x+k\\ y-k\leq j\leq y+k}} \sin\phi(i,j)\right)^2}}{(2k+1)^2} \tag{4.14}$$

$\phi(i,j)$ is the phase value of a pixel (i,j) in the $(2k+1)\times(2k+1)$ neighbourhood centred around pixel (x,y) where k is a positive integer. The pseudocorrelation maps do not always accurately depict the actual phase quality of the map (Lu et al. 2012). Pseudocorrelation maps will show areas with large phase variations as "low quality" regardless of the actual quality of the phase values, this is due to the fact that the pseudocorrelation values are based directly on the phase values (Ghiglia & Pritt 1998). It is for this reason the most reliable quality maps use phase derivatives instead.

4.4.3 Phase Derivative Variance

The most common approach to determining phase quality, (Lu et al. 2012, Suprijanto et al. 2001), is based on phase gradients. In the case of Phase Derivative Variance (PDV) maps, the quality of the phase is determined by the local variance between the partial derivatives of the phase data. The partial derivatives in this instance are the wrapped phase gradients of adjacent pixels along the x- and y-axis.
Extending Equation (4.7) to two dimensions gives us the two partial derivatives:

$$\begin{aligned}\Delta^x_{\mathrm{W}}\phi(x,y) &= \mathrm{W}\left[\phi(x+1,y)-\phi(x,y)\right]\\ \Delta^y_{\mathrm{W}}\phi(x,y) &= \mathrm{W}\left[\phi(x,y+1)-\phi(x,y)\right]\end{aligned} \tag{4.15}$$

Δ^x_{W} and Δ^y_{W} are thus the wrapped partial derivative operators along the x- and y-axis, respectively. Equation (4.15) is used to generate wrapped phase derivative maps which are used to create the actual phase quality map:

$$q_{\mathrm{pdv}}(x,y) = \frac{\sqrt{\sum_{\substack{x-k\leq i\leq x+k\\ y-k\leq j\leq y+k}} \left(\Delta^x_{\mathrm{W}}\phi(i,j) - \overline{\Delta^x_{\mathrm{W}}\phi(x,y)}\right)^2} + \sqrt{\sum_{\substack{x-k\leq i\leq x+k\\ y-k\leq j\leq y+k}} \left(\Delta^y_{\mathrm{W}}\phi(i,j) - \overline{\Delta^y_{\mathrm{W}}\phi(x,y)}\right)^2}}{(2k+1)^2} \tag{4.16}$$

$\overline{\Delta^x_{\mathrm{W}}\phi(x,y)}$ and $\overline{\Delta^y_{\mathrm{W}}\phi(x,y)}$ are the averages of their respective phase derivative values in the $(2k+1)\times(2k+1)$ window which is centred around the pixel (x,y) where k is a positive integer. By applying Equation (4.16) to the phase derivative maps, the PDV map can be generated. It is important to note that PDV maps actually provide an inverted measure of quality, that is low PDV values indicate high quality phase and high PDV values indicate low quality phase. Although PDV maps have been shown to be reliable methods of guiding unwrapping they still suffer from problems related to mischaracterisation of regions with high quality phase when the region has been under-sampled (Lu et al. 2007) and in regions of very low SNR it may incorrectly register some pixels as high quality by random chance.

Zhang et al. (2014) proposed the use of a median filter on the standard PDV maps to counter the problem of misregistration of noisy pixels. These "high quality" noisy pixels are generally found in isolation in an area of predominantly low quality phase. By applying a 2D median filter to the PDV map these errors can be avoided by suppressing the noisy pixels.

Weighted Phase Gradient

Lu et al. (2012) investigated the use of various weighting techniques to improve the reliability of the standard PDV maps. PDV maps are only sensitive to phase differences between adjacent pixels along the x- and y-axes, this makes it vulnerable to noise caused by other neighbouring pixels in the region. This vulnerability means that the PDV quality maps may not always be reliable. The weighted mask uses more of the neighbourhood surrounding the pixel under investigation to give a more reliable phase gradient value, called a Weighted Phase Gradient (WPG).

$$\begin{aligned}\Delta_w^x(x,y) &= \phi * w^x[x,y] \\ &= \sum_{\substack{-m\leq i\leq m \\ -n\leq j\leq n}} \phi(x-i,y-j)w^x(i,j) \\ \Delta_w^y(x,y) &= \phi * w^y[x,y] \\ &= \sum_{\substack{-m\leq i\leq m \\ -n\leq j\leq n}} \phi(x-i,y-j)w^y(i,j) \end{aligned} \tag{4.17}$$

The wrapped phase map ϕ of size (M, N) is convolved with weighted masks w^x and w^y, both of size $(2m+1, 2n+1)$, to produce the WPG maps Δ_w^x and Δ_w^y as shown in Equation (4.17). Each value is then wrapped using the wrapping operator W to give $\Delta_{w,\mathrm{W}}^x$ and $\Delta_{w,\mathrm{W}}^y$. Lu et al. (2012) makes use of several weighted masks, largely of the form shown below:

$$\underset{w^x}{\begin{pmatrix} w^x(-1,-1) & w^x(0,-1) & w^x(1,-1) \\ w^x(-1,0) & \underline{w^x(0,0)} & w^x(1,0) \\ w^x(-1,1) & w^x(0,1) & w^x(1,1) \end{pmatrix}} \quad \underset{w^y}{\begin{pmatrix} w^y(-1,-1) & w^y(0,-1) & w^y(1,-1) \\ w^y(-1,0) & \underline{w^y(0,0)} & w^y(1,0) \\ w^y(-1,1) & w^y(0,1) & w^y(1,1) \end{pmatrix}}$$

The underlined value in each array is the origin of the mask and is applied to each (x,y) position in ϕ. The 2×2 masks shown below can be used in Equation (4.17) to return a non-weighted PDV map.

$$\underset{w^x}{\begin{pmatrix} \underline{1} & -1 \\ 0 & 0 \end{pmatrix}} \quad \underset{w^y}{\begin{pmatrix} \underline{1} & 0 \\ -1 & 0 \end{pmatrix}}$$

The other weighted masks used are based on the Prewitt and Sobel operators. **A** and **B** are the Prewitt mask and diagonal Prewitt mask, respectively,

$$\begin{matrix} & \mathbf{A} & & & \mathbf{B} & \\ \begin{pmatrix} 1 & 0 & -1 \\ 1 & \underline{0} & -1 \\ 1 & 0 & -1 \end{pmatrix} & \begin{pmatrix} 1 & 1 & 1 \\ 0 & \underline{0} & 0 \\ -1 & -1 & -1 \end{pmatrix} & & \begin{pmatrix} 1 & 1 & 0 \\ 1 & \underline{0} & -1 \\ 0 & -1 & -1 \end{pmatrix} & \begin{pmatrix} 0 & -1 & -1 \\ 1 & \underline{0} & -1 \\ 1 & 1 & 0 \end{pmatrix} \\ w^x & w^y & & w^x & w^y \end{matrix}$$

while **C** and **D** are the Sobel mask and diagonal Sobel mask, respectively.

$$\begin{array}{cccc} & \mathbf{C} & & \mathbf{D} \\ \begin{pmatrix} 1 & 0 & -1 \\ 2 & \underline{0} & -2 \\ 1 & 0 & -1 \end{pmatrix} & \begin{pmatrix} 1 & 2 & 1 \\ 0 & \underline{0} & 0 \\ -1 & -2 & -1 \end{pmatrix} & \begin{pmatrix} 2 & 1 & 0 \\ 1 & \underline{0} & -1 \\ 0 & -1 & -2 \end{pmatrix} & \begin{pmatrix} 0 & -1 & -2 \\ 1 & \underline{0} & -1 \\ 2 & 1 & 0 \end{pmatrix} \\ w^x & w^y & w^x & w^y \end{array}$$

The WPG quality map is generated by subbing $\Delta^x_{w,\mathrm{W}}$ and $\Delta^y_{w,\mathrm{W}}$ into Equation (4.16)

$$q_{\mathrm{wpg}}(x,y) = \frac{\sqrt{\sum_{\substack{x-k\leq i\leq x+k \\ y-k\leq j\leq y+k}} \left(\Delta^x_{w,\mathrm{W}}\phi(i,j) - \overline{\Delta^x_{w,\mathrm{W}}\phi(x,y)}\right)^2} + \sqrt{\sum_{\substack{x-k\leq i\leq x+k \\ y-k\leq j\leq y+k}} \left(\Delta^y_{w,\mathrm{W}}\phi(i,j) - \overline{\Delta^y_{w,\mathrm{W}}\phi(x,y)}\right)^2}}{(2k+1)^2} \tag{4.18}$$

4.5 Validation of Techniques

The use of the Goldstein branch cut and PDV QGPU was primarily based on its popular use in other texts (Ghiglia & Pritt 1998, Lu et al. 2012, Spottiswoode 2006, Suprijanto et al. 2001, Ying 2006, Zhang et al. 2014). Temporal phase unwrapping is a simple but powerful unwrapping tool (Huntley & Saldner 1993) and as such was used as an unwrapping technique in this work. The 3D branch cut method was used to overcome the problems related to noise to which the other methods were susceptible. These methods were validated using simulated datasets.

To simulate a blood vessel as it is found in a standard 2D PC-MRA phase dataset, the MATLAB function *fit* was used to fit volunteer blood flow data to a five term Fourier transform. This was done to produce a simulated blood flow curve that could have its parameters altered. The flow curve could then be down-sampled to simulate the number of frames in a normal PC-MRA acquisition. The flow curve was then replicated and tiled to create a time resolved 2D image, with the flow along the time axis. The diameter of the ROI could be changed to simulate different vessel diameters. The ROI was then altered spatially so that the outermost pixels had a near zero value and the innermost pixels had the maximum flow value, this simulated laminar flow. Random Gaussian noise was added to the image to simulate the inherent noise present in

PC-MRA acquisitions. The variance of the noise was altered to change the noise density to simulate different SNR values. Finally by altering the v_{enc} parameter, the data could be wrapped at any chosen flow value.

To simulate a 3D dataset, one frame of a 4D PC-MRA acquisition, a volume was created with values varying linearly from zero to a multiple of the v_{enc} parameter, called the scaling factor. Random Gaussian noise was added to the image; the variance could be altered to change the noise density in the volume. The volume was then wrapped using the v_{enc} parameter.

The respective techniques were then used to unwrap the simulated datasets. Using a method described by Ghiglia & Pritt (1998), the unwrapped phase map, ϕ', was compared against the wrapped phase gradients Δ^x_W and Δ^y_W, defined in Section 4.4.3. The method uses a weighted L^p-norm to determine how closely the phase gradient of ϕ' matches the wrapped phase gradients, this is shown in Equation (4.19).

$$\begin{aligned}\epsilon =& \frac{1}{MN}\sum_{i=0}^{M-2}\sum_{j=0}^{N-1} q^x(i,j)\ |\phi'(i+1,j)-\phi'(i,j)-\Delta^x_W(i,j)|^p \\ &+ \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-2} q^y(i,j)\ |\phi'(i,j+1)-\phi'(i,j)-\Delta^y_W(i,j)|^p \end{aligned} \tag{4.19}$$

ϵ is the error value that is used as a measure of success. q^x and q^y are user defined weights for the x- and y-direction, respectively. These weights can be derived from the PDV quality map, defined in Section 4.4.3, as follows:

$$q^x(i,j) = \min(q_{\text{pdv}}(i+1,j), q_{\text{pdv}}(i,j))$$

$$q^y(i,j) = \min(q_{\text{pdv}}(i,j+1), q_{\text{pdv}}(i,j))$$

M and N are the number of columns and rows in ϕ', respectively. The L_p-norm is performed for $p = 0, 1, 2$ where for $p = 0$ the solution is the number of samples where the gradient of the unwrapped solution and the wrapped map do not match. For $p = 1$ the solution is the average absolute deviation of the gradients, and for $p = 2$ the solution is the average squared absolute deviation of the gradients.

Relatively large values of ϵ indicate a bad "fit" between ϕ' and the wrapped phase

gradients meaning the unwrapping operation has likely failed. However, ϵ is not an absolute measure of success but rather needs to be interpreted by the user to determine if the operation was a success or not. This was done visually by comparing ϕ' with the true unwrapped phase map, Φ. Results of the validation for each technique are presented in Chapter 5.

Chapter 5

Results

The phase unwrapping techniques used to unwrap the wrapped PC-MRA datasets were temporal phase unwrapping, the Goldstein-Zebker-Werner branch cut algorithm, quality guided phase unwrapping, and the 3D branch cut method, all previously shown in Chapter 4. Another technique used, but not previously mentioned, incorporated facets of both the temporal phase unwrapping and Goldstein branch cut algorithms.
All of these algorithms were validated using simulated datasets and then applied to the PC-MRA datasets containing phase wrapping in the ROIs, that is the major blood vessels of the upper arm for the volunteers and the haemodialysis access vessels for the patients. The results are given below and are discussed in Chapter 6.

5.1 Validation

5.1.1 Temporal Phase Unwrapping

Temporal phase unwrapping, described in Section 4.1, was tested using the validation technique described previously in Section 4.5. Several flow curves were simulated with an average peak velocity of 65.6±10.3 cm/s. Each frame of the image was 50×50 pixels and the diameter of the ROI was 10 pixels. The v_{enc} parameter, the number of images, and the noise variance were varied to simulate several different kinds of acquisitions. The diameter of the ROI was not varied as temporal phase unwrapping is affected primarily by changes along the temporal axis meaning spatial changes have little to no effect.

A threshold was used so that if more than 10% of the pixels in the region of interest were found to be corrupted, that is it differed by more than 2π from the true pixel value, the operation was considered to have failed. It is important to note that a failure in this sense does not mean that the algorithm itself failed but rather that the data differed from the true phase map too greatly to be considered usable.

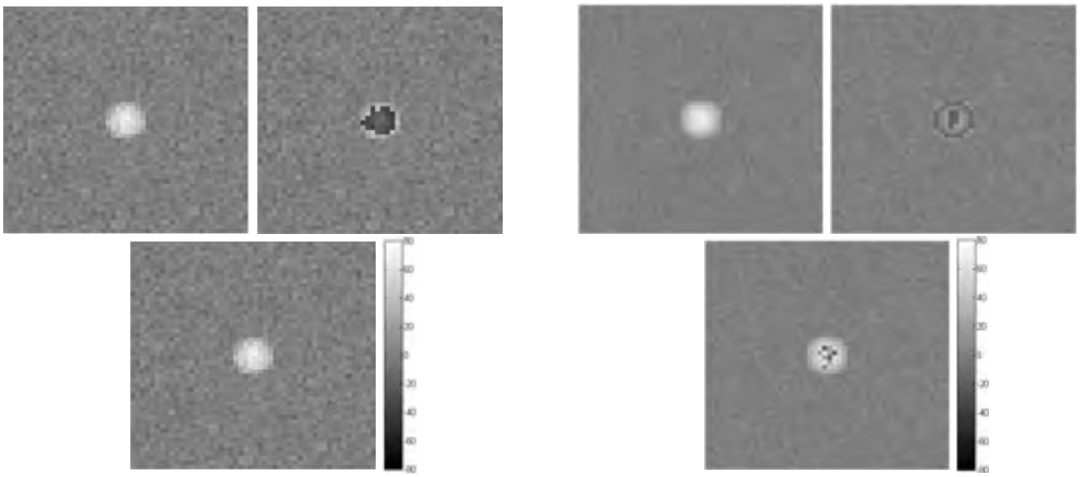
Table 5.1 gives the parameters used for each test and the results from the L_p-norm evaluation. The L_p-norm evaluation shows the maximum, mean, and standard deviation of the L_p values from all the frames. Each test was run multiple times with different blood flow curves. The average result is given for each series of tests for brevity except in the case where the result varied between pass and fail. In Figures 5.1a and 5.1b the results of two simulations are shown. Figure 5.1a shows a correctly unwrapped image; the v_{enc} was set at 40 cm/s, the noise variance was 0.005, and there were 25 images. Figure 5.1b shows an incorrectly unwrapped image; the v_{enc} was set at 20 cm/s, the noise variance was 0.005, and there were 25 images.

Table 5.1: Validation of Temporal Phase Unwrapping

	№ of Images	v_{enc} [cm/s]	Variance	L_1 Max (mean±sd)	L_2 Max (mean±sd)	Pass/Fail
1	25	60	0.005	0.000(0.000±0.000)	0.000(0.000±0.000)	Pass
2	25	40	0.005	0.005(0.003±0.002)	0.038(0.019±0.012)	Pass
3	25	20	0.005	0.099(0.066±0.030)	0.817(0.533±0.255)	Fail
4	25	30	0.005	0.017(0.009±0.005)	0.121(0.066±0.040)	Pass
5	29	20	0.005	0.067(0.042±0.021)	0.512(0.318±0.161)	Fail
6	50	20	0.005	0.021(0.004±0.005)	0.135(0.024±0.029)	Pass
7	25	30	0.01	0.072(0.048±0.023)	0.493(0.323±0.159)	Fail
8	25	30	0.01	0.062(0.038±0.019)	0.415(0.258±0.133)	Pass
9	25	30	0.01	0.075(0.047±0.024)	0.511(0.316±0.160)	Fail
10	25	30	0.01	0.031(0.017±0.009)	0.192(0.109±0.056)	Pass
11	25	30	0.01	0.094(0.053±0.027)	0.709(0.392±0.210)	Fail
12	25	30	0.02	0.852(0.452±0.252)	5.878(3.031±1.744)	Fail
13	25	40	0.02	0.804(0.408±0.232)	5.422(2.685±1.560)	Fail
14	25	60	0.02	0.811(0.384±0.244)	5.466(2.499±1.618)	Fail
15	25	60	0.02	0.850(0.403±0.247)	5.633(2.615±1.632)	Pass
16	25	60	0.02	0.802(0.393±0.246)	5.633(2.630±1.713)	Pass
17	25	60	0.02	0.930(0.441±0.268)	6.477(3.023±1.871)	Pass
18	25	60	0.02	0.718(0.389±0.214)	4.903(2.590±1.486)	Fail

5.1.2 Goldstein Branch Cut Method

The Goldstein branch cut method, described in Section 4.2.4, was tested using the validation technique described previously in Section 4.5. Several flow curves were simulated with an average peak velocity of 64.3±11.0 cm/s. Each set of images was composed of 25 images, each with a size of 50 × 50 pixels. The v_{enc} parameter, and the noise



(a) Successfully unwrapped image

(b) Incorrectly unwrapped image

Figure 5.1: Examples of temporally unwrapped phase images. Clockwise from top left, for both: True phase, wrapped phase, and unwrapped phase. The grey levels are scaled between -80 and 80 cm/s for all images.

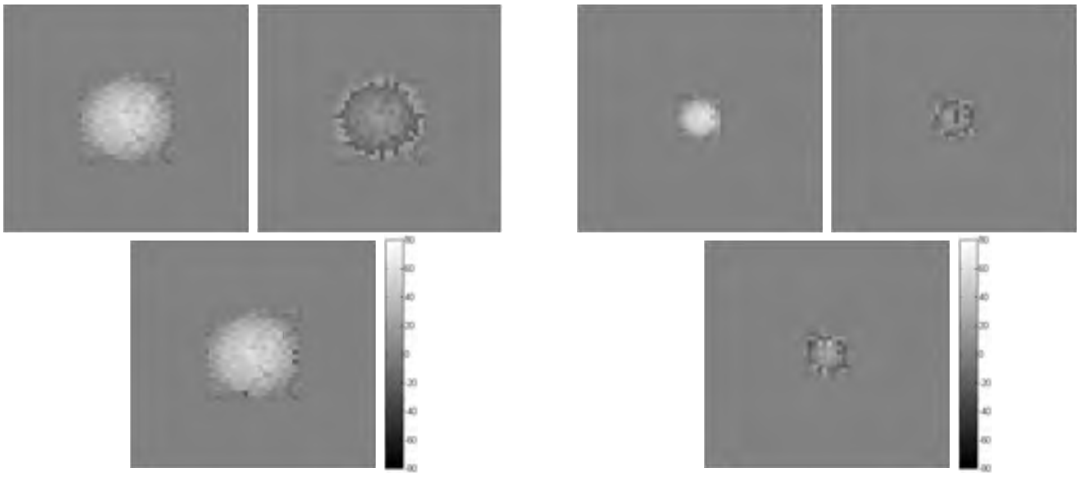
variance were varied to simulate several different kinds of acquisitions. The diameter of the region of interest was also varied to simulate various blood vessel sizes. The number of images was kept constant as the Goldstein branch cut method is a spatial method of unwrapping and so changes along the temporal axis will not affect the result.

A threshold was used so that if more than 10% of the pixels in the region of interest were found to be corrupted, that is it differed by more than 2π from the true pixel value, the operation was considered to have failed and as such the image was not usable.

Table 5.2 gives the parameters used for each test and the results from the L_p-norm evaluation. The L_p-norm evaluation shows the maximum, mean, and standard deviation of the L_p values from all the frames. Each test was run multiple times with different blood flow curves. The average result is given for each series of tests for brevity except in the case where the result varied between pass and fail. In Figures 5.2a and 5.2b the results of two simulations are shown. Figure 5.2a shows a successfully unwrapped image which matches the true phase image very closely; the v_{enc} was set at 20 cm/s, the noise variance was 0.02, and the diameter of the ROI was 20 pixels. Figure 5.2b shows an incorrectly unwrapped image due the combination of noise and small ROI diameter. The v_{enc} was set at 20 cm/s, the noise variance was 0.02, and the diameter of the ROI was 10 pixels.

Table 5.2: Validation of the Goldstein-Zebker-Werner Branch Cut Method

	Diameter [px]	v_{enc} **[cm/s]**	**Variance**	**№ of Residues**	**L_1 Max (mean±sd)**	**L_2 Max (mean±sd)**	**Pass/Fail**
1	10	60	0.005	0	0.000(0.000±0.000)	0.000(0.000±0.000)	Pass
2	10	40	0.005	0	0.000(0.000±0.000)	0.000(0.000±0.000)	Pass
3	10	20	0.005	22	0.005(0.000±0.001)	0.033(0.002±0.007)	Pass
4	10	20	0.005	34	0.007(0.000±0.002)	0.044(0.003±0.010)	Pass
5	10	20	0.005	40	0.010(0.001±0.002)	0.064(0.005±0.015)	Fail
6	10	20	0.005	30	0.006(0.000±0.001)	0.037(0.003±0.009)	Pass
7	10	20	0.005	14	0.002(0.000±0.001)	0.012(0.001±0.003)	Pass
8	8	20	0.005	39	0.006(0.001±0.002)	0.037(0.003±0.010)	Fail
9	8	40	0.005	1	0.000(0.000±0.000)	0.001(0.000±0.000)	Pass
10	8	40	0.01	14	0.002(0.000±0.000)	0.012(0.001±0.002)	Pass
11	8	40	0.02	90	0.005(0.001±0.001)	0.029(0.005±0.007)	Pass
12	8	40	0.05	358	0.010(0.005±0.002)	0.064(0.030±0.014)	Fail
13	20	20	0.005	2	0.000(0.000±0.000)	0.001(0.000±0.000)	Pass
14	20	20	0.01	15	0.001(0.000±0.000)	0.009(0.001±0.002)	Pass
15	20	20	0.02	201	0.010(0.002±0.002)	0.063(0.014±0.015)	Pass
16	20	20	0.05	1626	0.056(0.028±0.010)	0.410(0.182±0.075)	Fail
17	20	20	0.05	1754	0.054(0.029±0.009)	0.338(0.181±0.056)	Fail
18	20	20	0.05	1704	0.046(0.027±0.007)	0.309(0.174±0.045)	Pass
19	20	20	0.05	1682	0.067(0.033±0.014)	0.510(0.216±0.104)	Fail
20	20	20	0.05	1668	0.041(0.028±0.006)	0.256(0.177±0.043)	Pass
21	10	20	0.02	208	0.015(0.003±0.004)	0.101(0.020±0.029)	Fail
22	10	20	0.02	154	0.009(0.002±0.003)	0.073(0.012±0.018)	Fail
23	10	20	0.02	104	0.006(0.001±0.002)	0.041(0.007±0.012)	Fail
24	10	20	0.02	144	0.012(0.002±0.003)	0.076(0.012±0.017)	Pass
25	10	20	0.02	202	0.010(0.002±0.003)	0.065(0.015±0.017)	Fail



(a) Successfully unwrapped image

(b) Incorrectly unwrapped image

Figure 5.2: Examples of unwrapped phase images using the Goldstein branch cut method. Clockwise from top left, for both: True phase, wrapped phase, and unwrapped phase. The grey levels are scaled between -80 and 80 cm/s for all images.

5.1.3 Quality Guided Phase Unwrapping

The PDV-QGPU method, described in Section 4.4.3, was tested using the validation technique described previously in Section 4.5. Several flow curves were simulated with an average peak velocity of 63.2±7.5 cm/s. Each set of images was composed of 25 images, each with a size of 50×50 pixels. The v_{enc} parameter, and the noise variance were varied to simulate several different kinds of acquisitions. The width of the region of interest was also varied to simulate various blood vessel sizes. The number of images was kept constant as quality guided phase unwrapping is a spatial technique and so changes along the temporal axis will not affect the result.

A threshold was used so that if more than 10% of the pixels in the region of interest were found to be corrupted, that is it differed by more than 2π from the true pixel value, the operation was considered to have failed and the image was considered unusable.

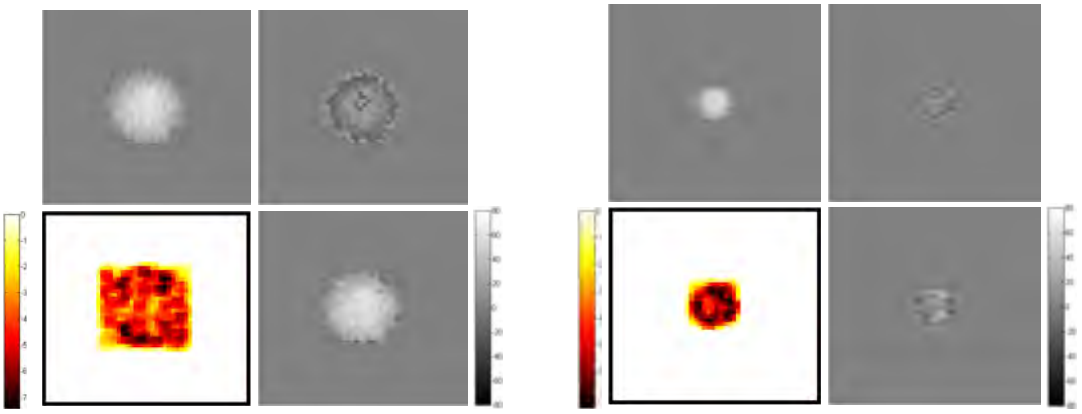
Table 5.3 gives the parameters used for each test and the results from the L_p-norm evaluation. The L_p-norm evaluation shows the maximum, mean, and standard deviation of the L_p values from all the frames. Each test was run multiple times with different blood flow curves. The average result is given for each series of tests for brevity except in the case where the result varied between pass and fail. In Figures 5.3a and 5.3b the results of two simulations are shown. Figure 5.3a shows a successfully unwrapped image which matches the true phase image very closely; the v_{enc} was set at 20 cm/s, the noise variance was 0.02, and the diameter of the ROI was 20 pixels. Figure 5.3b shows an incorrectly unwrapped image due the combination of noise and small ROI diameter. The v_{enc} was set at 20 cm/s, the noise variance was 0.02, and the diameter of the ROI was 10 pixels.

5.1.4 3D Branch Cut Method

The 3D branch cut method was tested using the validation for 3D volumes previously discussed in Section 4.5. Several volumes were generated with various matrix sizes, v_{enc} scaling factor, and noise variance. The velocity scaling and matrix size made it possible to reduce the distance between contiguous phase wraps and as such simulate rapidly changing blood flow along spatial dimensions. The v_{enc} was set at 60 cm/s for all volumes as, for this set of simulations, the actual velocity is not as important for testing as the velocity scaling factor.

Table 5.3: Validation of PDV QGPU

	Diameter [px]	v_{enc} **[cm/s]**	**Variance**	**L_1 Max (mean±sd)**	**L_2 Max (mean±sd)**	**Pass/Fail**
1	10	60	0.005	0.000 (0.000±0.000)	0.000 (0.000±0.000)	Pass
2	10	40	0.005	0.000 (0.000±0.000)	0.000 (0.000±0.000)	Pass
3	10	20	0.005	0.010 (0.001±0.002)	0.073 (0.005±0.018)	Pass
4	10	20	0.005	0.017 (0.001±0.004)	0.145 (0.009±0.032)	Fail
5	10	20	0.005	0.004 (0.000±0.001)	0.022 (0.001±0.004)	Pass
6	10	20	0.005	0.011 (0.001±0.003)	0.075 (0.005±0.017)	Pass
7	10	20	0.005	0.004 (0.000±0.001)	0.027 (0.001±0.005)	Pass
8	8	20	0.005	0.006 (0.001±0.002)	0.045 (0.004±0.011)	Fail
9	8	40	0.005	0.000 (0.000±0.000)	0.000 (0.000±0.000)	Pass
10	8	40	0.01	0.001 (0.000±0.000)	0.004 (0.000±0.001)	Pass
11	8	40	0.02	0.002 (0.000±0.001)	0.015 (0.002±0.004)	Pass
12	8	40	0.05	0.009 (0.004±0.002)	0.060 (0.023±0.015)	Fail
13	20	20	0.01	0.001 (0.000±0.000)	0.005 (0.000±0.001)	Pass
14	20	20	0.02	0.014 (0.003±0.003)	0.090 (0.017±0.020)	Pass
15	20	20	0.05	0.077 (0.040±0.015)	0.696 (0.280±0.134)	Fail
16	20	20	0.05	0.063 (0.037±0.013)	0.486 (0.257±0.105)	Pass
17	20	20	0.05	0.081 (0.043±0.015)	0.577 (0.297±0.114)	Fail
18	20	20	0.05	0.069 (0.038±0.014)	0.558 (0.259±0.105)	Fail
19	20	20	0.05	0.074 (0.039±0.013)	0.529 (0.266±0.103)	Fail
20	10	20	0.02	0.010 (0.001±0.002)	0.064 (0.006±0.015)	Fail
21	10	20	0.02	0.010 (0.001±0.003)	0.064 (0.007±0.017)	Pass
22	10	20	0.02	0.011 (0.002±0.003)	0.072 (0.012±0.021)	Pass
23	10	20	0.02	0.012 (0.001±0.003)	0.084 (0.008±0.020)	Fail
24	10	20	0.02	0.015 (0.002±0.004)	0.104 (0.011±0.025)	Fail
25	10	20	0.02	0.014 (0.001±0.003)	0.105 (0.009±0.022)	Fail



(a) Successfully unwrapped image

(b) Incorrectly unwrapped image

Figure 5.3: Examples of QGPU using a PDV map. Clockwise from top left, for both: True phase, wrapped phase, unwrapped phase, and quality map. The grey levels are scaled between -80 and 80 cm/s for all phase images.

A threshold was used so that if more than 10% of the pixels in the region of interest were found to be corrupted, that is it differed by more than 2π from the true pixel value, the operation was considered to have failed and the image was considered unusable. The volumes were also compared directly against the true, unwrapped phase volume.

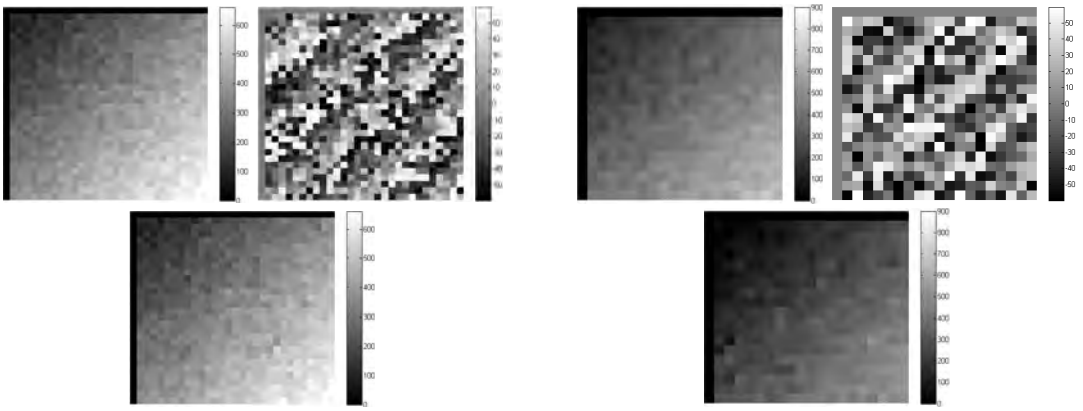
Table 5.4 gives the parameters used for each test and the results from the L_p-norm evaluation. Each test was run multiple times with different phase volumes. The average result is given for each series of tests for brevity except in the case where the result varied between pass and fail. For each test with only one entry the number of residues, number of PSLs, and L_p-norm values represent the average values and all had the same pass or failure result. In Figures 5.4a and 5.4b the results of two simulations are shown. Figure 5.4a shows a successfully unwrapped image which matches the true phase image very closely; the v_{enc} scaling factor was set at 15, the noise variance was 0.05, and the size of the volume was $30 \times 30 \times 30$ pixels. Figure 5.4b shows an incorrectly unwrapped image due the combination of noise and the small size of the volume. The v_{enc} scaling factor was set at 15, the noise variance was 0.05, and the size of the volume was $20 \times 20 \times 20$ pixels.

Table 5.4: Validation of 3D branch cut method

	Volume Dimensions [px]	Scaling Factor	Variance	№ of Resiudes	№ of PSLs	L_1	L_2	Pass/Fail
1	50×50×50	5	0.005	0	0	0.000	0.000	Pass
2	50×50×50	5	0.01	69	18	0.000	0.001	Pass
3	50×50×50	5	0.02	5137	1184	0.021	0.134	Pass
4	50×50×50	5	0.05	54386	9389	0.290	1.821	Pass
5	30×30×30	5	0.05	10924	1982	0.252	1.586	Pass
6	30×30×30	8	0.05	11301	2002	0.323	2.029	Pass
7	30×30×30	10	0.05	11503	1981	0.302	1.897	Pass
8	30×30×30	15	0.05	12677	2095	0.323	2.031	Pass
9	20×20×20	15	0.005	2	1	0.000	0.001	Pass
10	20×20×20	15	0.01	86	23	0.005	0.033	Pass
11	20×20×20	15	0.02	984	232	0.075	0.474	Pass
12	20×20×20	15	0.05	3688	604	0.392	2.494	Fail
13	20×20×20	15	0.05	3881	663	0.320	2.017	Fail
14	20×20×20	15	0.05	3881	628	0.360	2.276	Fail
15	20×20×20	15	0.05	3883	624	0.395	2.493	Pass

5.2 PC-MRA Data

Temporal phase unwrapping, the Goldstein-Zebker-Werner branch cut algorithm, and QGPU were applied to the 2D PC-MRA datasets only while the 3D branch cut method was applied to the 4D PC-MRA datasets only.



(a) Successfully unwrapped image

(b) Incorrectly unwrapped image

Figure 5.4: Examples of unwrapped images using 3D branch cut method. Clockwise from top left, for both: True phase, wrapped phase, unwrapped phase. The grey levels are shown to the left of each image.

The magnitude images were used to define a mask around the perimeter of each volunteer's upper arm. This was done to remove the noisy air cavity surrounding the arm. The mask was then applied to the phase images, prior to the application of the temporal unwrapping. These masked phase images were used during the L_p-norm evaluation.
Determining whether the dataset was successfully unwrapped was completed by visual inspection. Unlike with the validation, there is no true phase dataset to compare the unwrapped results against. For this reason the unwrapped datasets were compared against other volunteers with little to no phase wrapping as these were considered the gold standard. If the ROIs matched the blood flow profile and the flow velocity fell within the expected range of values for the particular vessel, the dataset was considered to have been successfully unwrapped.

5.2.1 Temporal Phase Unwrapping

Temporal phase unwrapping was applied to the 2D PC-MRA volunteer data only as it was found to be unsuitable, by itself, for the majority of the patient data.
The results are shown below in Table 5.5. The ROI Diameter field gives the approximate diameter, in pixels, of the vessel under investigation. In this case the vessel was always the brachial artery as the cephalic vein never contained any phase wrapping. The L_p-norm evaluation shows the maximum, mean, and standard deviation of the L_p values from all the frames of the dataset.

Table 5.5: Temporal Phase Unwrapping of 2D PC-MRA

Patient ID	Image Size	№ of Images	v_{enc} [cm/s]	ROI Diameter [px]	Image Quality	L_1 Max (mean±sd)	L_2 Max (mean±sd)	Pass/Fail
Volunteer 1	192×256	12	15	7	High	0.253(0.065±0.087)	1.772(0.440±0.602)	Fail
Volunteer 1	192×256	22	30	7	High	0.109(0.026±0.331)	0.698(0.165±0.212)	Fail
Volunteer 1	144×192	22	40	7	High	0.015(0.003±0.004)	0.092(0.019±0.027)	Pass
Volunteer 2	192×256	9	20	7	High	0.003(0.001±0.001)	0.019(0.007±0.007)	Fail
Volunteer 3	192×256	9	20	7	High	0.006(0.003±0.002)	0.037(0.017±0.013)	Fail
Volunteer 4	192×256	12	20	9	Low	1.609(0.783±0.525)	13.403(5.894±4.352)	Fail
Volunteer 4	192×256	12	20	9	High	0.097(0.032±0.035)	0.633(0.204±0.224)	Fail
Volunteer 4	192×256	14	80	9	Medium	0.235(0.081±0.079)	1.392(0.522±0.516)	Pass
Volunteer 4	192×256	14	80	9	High	0.177(0.063±0.057)	1.145(0.403±0.369)	Fail

5.2.2 Goldstein Branch Cut Method

The Goldstein branch cut method was applied to the 2D PC-MRA volunteer and patient data.

The results are shown below in Table 5.6. The ROI Diameter field gives the approximate diameter, in pixels, of the vessel under investigation. For the volunteers the vessel under investigation was always only the brachial artery due to the presence of phase wrapping. Depending on the location of the scan in the patients' upper arms, above, below, or through the anastomosis, would determine the number of ROIs. In general the brachial artery was always visible while the fistula or graft was only visible for locations above and through the anastomosis. For these specific entries a second width value is given that represents that vessel's diameter. The Image Quality field was closely related to, but not solely determined by, the number of residues in the image. The L_p-norm evaluation shows the maximum, mean, and standard deviation of the L_p values from all the frames of the dataset.

Table 5.6: The Goldstein Branch Cut Method on 2D PC-MRA

Patient ID	Image Size	№ of Images	v_{enc} [cm/s]	ROI Diameter [px]	Image Quality	№ of Residues	L_1 Max (mean±sd)	L_2 Max (mean±sd)	Pass/Fail
Volunteer 1	192×256	12	15	7	High	4127	0.002(0.001±0.001)	0.014(0.003±0.005)	Fail
Volunteer 1	192×256	22	30	7	High	429	0.001(0.000±0.000)	0.004(0.001±0.001)	Fail
Volunteer 1	144×192	22	40	7	High	40	0.000(0.000±0.000)	0.002(0.000±0.001)	Pass
Volunteer 2	192×256	9	20	7	High	12	0.000(0.000±0.000)	0.001(0.000±0.000)	Fail
Volunteer 3	192×256	9	20	7	High	14	0.001(0.000±0.000)	0.004(0.000±0.001)	Fail
Volunteer 4	192×256	12	20	9	Low	15791	0.040(0.025±0.013)	0.259(0.162±0.086)	Fail
Volunteer 4	192×256	12	20	9	High	390	0.002(0.000±0.001)	0.011(0.003±0.003)	Fail
Volunteer 4	192×256	14	80	9	Medium	1394	0.010(0.004±0.003)	0.063(0.025±0.021)	Pass
Volunteer 4	192×256	14	80	9	High	1864	0.006(0.003±0.001)	0.036(0.020±0.010)	Pass
Fistula 1	192×256	14	80	12;25[a]	Medium	9694	0.013(0.008±0.004)	0.084(0.047±0.028)	Pass
Fistula 1	192×256	14	80	20	Low	28093	0.006(0.005±0.001)	0.038(0.030±0.005)	Pass
Fistula 2	208×256	14	80	12;40[a]	High	12286	0.006(0.004±0.002)	0.038(0.021±0.011)	Fail
Fistula 2	192×256	14	80	15	High	124	0.001(0.001±0.000)	0.007(0.002±0.002)	Pass
Fistula 3	192×256	14	80	10;35[a]	Medium	11462	0.010(0.005±0.003)	0.062(0.031±0.017)	Fail
Fistula 3	192×256	14	80	20	High	4868	0.005(0.002±0.001)	0.033(0.012±0.008)	Fail
Fistula 4	192×256	14	80	10;22[a]	High	5352	0.014(0.008±0.005)	0.090(0.049±0.035)	Pass
Fistula 4	192×256	14	80	10	High	438	0.003(0.002±0.001)	0.022(0.010±0.007)	Fail
Graft 1	192×256	14	80	10;10[b]	Low	152	0.000(0.000±0.000)	0.003(0.001±0.001)	Fail
Graft 1	192×256	14	80	10;8[b]	High	258	0.001(0.001±0.000)	0.007(0.004±0.002)	Fail
Graft 2	192×256	10	80	12;8[b]	Low	18816	0.027(0.020±0.008)	0.174(0.124±0.050)	Fail
Graft 2	192×256	10	80	10	High	2453	0.000(0.000±0.000)	0.001(0.000±0.000)	Pass
Graft 3	192×256	14	80	11;14[b]	Medium	7223	0.010(0.006±0.003)	0.058(0.038±0.017)	Fail
Graft 3	192×256	14	80	12	High	394	0.001(0.001±0.000)	0.006(0.003±0.002)	Fail

[a] The second value gives the width of the cephalic vein

[b] The second value gives the width of the AV graft

5.2.3 Spatio-temporal Phase Unwrapping

The Goldstein branch cut method and temporal phase unwrapping were combined together to provide a new technique for unwrapping the 2D PC-MRA datasets. This spatio-temporal phase unwrapping method was applied to the 2D PC-MRA volunteer and patient data.

The results are shown below in Table 5.7. The ROI Diameter field gives the approximate diameter, in pixels, of the vessel under investigation. For the volunteers the vessel under investigation was always only the brachial artery due to the presence of phase wrapping. Depending on the location of the scan in the patients' upper arms, above, below, or through the anastomosis, would determine the number of ROIs. In general the brachial artery was always visible while the fistula or graft was only visible for locations above and through the anastomosis. For these specific entries a second width value is given that represents that vessel's diameter. The Image Quality field was closely related to, but not solely determined by, the number of residues in the image. The L_p-norm evaluation shows the maximum, mean, and standard deviation of the L_p values from all the frames of the dataset.

Table 5.7: Spatio-temporal Phase Unwrapping of 2D PC-MRA

Patient ID	Image Size	№ of Images	v_{enc} [cm/s]	ROI Diameter [px]	Image Quality	L_1 Max (mean±sd)	L_2 Max (mean±sd)	Pass/Fail
Volunteer 1	192×256	12	15	7	High	0.017(0.004±0.005)	0.112(0.027±0.035)	Fail
Volunteer 1	192×256	22	30	7	High	0.011(0.004±0.003)	0.073(0.026±0.021)	Fail
Volunteer 1	144×192	22	40	7	High	0.002(0.001±0.001)	0.013(0.005±0.004)	Pass
Volunteer 2	192×256	9	20	7	High	0.009(0.007±0.002)	0.010(0.005±0.003)	Fail
Volunteer 3	192×256	9	20	7	High	0.019(0.015±0.004)	0.018(0.010±0.007)	Fail
Volunteer 4	192×256	12	20	9	Low	0.312(0.155±0.107)	2.396(0.988±0.873)	Fail
Volunteer 4	192×256	12	20	9	High	0.025(0.016±0.005)	0.085(0.032±0.028)	Fail
Volunteer 4	192×256	14	80	9	Medium	0.060(0.021±0.021)	0.386(0.132±0.133)	Pass
Volunteer 4	192×256	14	80	9	High	0.075(0.046±0.017)	0.417(0.202±0.123)	Fail
Fistula 1	192×256	14	80	12;25[a]	Medium	0.129(0.060±0.036)	0.954(0.353±0.296)	Pass
Fistula 1	192×256	14	80	20	Medium	0.126(0.085±0.035)	0.793(0.438±0.261)	Pass
Fistula 2	208×256	14	80	12;40[a]	High	0.074(0.034±0.022)	0.520(0.198±0.169)	Pass
Fistula 2	192×256	14	80	15	High	0.012(0.008±0.003)	0.049(0.029±0.019)	Pass
Fistula 3	192×256	14	80	10;35[a]	Medium	0.148(0.084±0.045)	0.980(0.451±0.328)	Pass
Fistula 3	192×256	14	80	20	High	0.072(0.045±0.020)	0.411(0.194±0.143)	Fail
Fistula 4	192×256	14	80	10;22[a]	High	0.184(0.081±0.061)	1.347(0.490±0.467)	Pass
Fistula 4	192×256	14	80	10	High	0.029(0.015±0.008)	0.153(0.056±0.051)	Fail
Graft 1	192×256	14	80	10;10[b]	Low	0.080(0.072±0.009)	0.163(0.082±0.043)	Fail
Graft 1	192×256	14	80	10;8[b]	High	0.012(0.010±0.002)	0.040(0.025±0.013)	Pass
Graft 2	192×256	10	80	12;8[b]	Low	0.401(0.227±0.127)	3.113(1.491±1.077)	Fail
Graft 2	192×256	10	80	10	High	0.010(0.006±0.002)	0.032(0.013±0.011)	Pass
Graft 3	192×256	14	80	11;14[b]	Medium	0.109(0.058±0.033)	0.807(0.347±0.271)	Fail
Graft 3	192×256	14	80	12	High	0.010(0.007±0.002)	0.034(0.017±0.010)	Fail

[a] The second value gives the width of the cephalic vein

[b] The second value gives the width of the AV graft

5.2.4 Quality Guided Phase Unwrapping

QGPU was applied to the 2D PC-MRA volunteer and patient data.

The results are shown below in Table 5.8. The ROI Diameter field gives the approximate

diameter, in pixels, of the vessel under investigation. For the volunteers the vessel under investigation was always only the brachial artery due to the presence of phase wrapping. The location of the scan in the patients' upper arms, above, below, or through the anastomosis, would determine the number of ROIs. In general the brachial artery was always visible while the fistula or graft was only visible for locations above and through the anastomosis. For these specific entries a second width value is given that represents that vessel's diameter. The L_p-norm evaluation shows the maximum, mean, and standard deviation of the L_p values from all the frames of the dataset.

Table 5.8: QGPU of 2D PC-MRA

Patient ID	Image Size	№ of Images	v_{enc} [cm/s]	ROI Diameter [px]	Image Quality	L_1 Max (mean±sd)	L_2 Max (mean±sd)	Pass/Fail
Volunteer 1	192×256	12	15	7	High	0.003(0.001±0.001)	0.017(0.003±0.005)	Fail
Volunteer 1	192×256	22	30	7	High	0.000(0.000±0.000)	0.002(0.001±0.001)	Fail
Volunteer 1	144×192	22	40	7	High	0.000(0.000±0.000)	0.000(0.000±0.000)	Pass
Volunteer 2	192×256	9	20	7	High	0.000(0.000±0.000)	0.000(0.000±0.000)	Fail
Volunteer 3	192×256	9	20	7	High	0.001(0.000±0.000)	0.004(0.000±0.001)	Fail
Volunteer 4	192×256	12	20	9	Low	0.039(0.023±0.014)	0.256(0.152±0.090)	Fail
Volunteer 4	192×256	12	20	9	High	0.001(0.000±0.000)	0.004(0.001±0.001)	Fail
Volunteer 4	192×256	14	80	9	Medium	0.007(0.003±0.003)	0.043(0.019±0.016)	Pass
Volunteer 4	192×256	14	80	9	High	0.006(0.003±0.001)	0.037(0.017±0.009)	Pass
Fistula 1	192×256	14	80	12;25[a]	Medium	0.011(0.006±0.003)	0.070(0.038±0.019)	Fail
Fistula 1	192×256	14	80	20	Medium	0.007(0.005±0.001)	0.047(0.034±0.008)	Pass
Fistula 2	208×256	14	80	12;40[a]	High	0.005(0.003±0.002)	0.032(0.019±0.010)	Fail
Fistula 2	192×256	14	80	15	High	0.001(0.000±0.000)	0.006(0.002±0.002)	Pass
Fistula 3	192×256	14	80	10;35[a]	Medium	0.006(0.004±0.002)	0.040(0.023±0.012)	Fail
Fistula 3	192×256	14	80	20	High	0.004(0.002±0.001)	0.025(0.010±0.006)	Fail
Fistula 4	192×256	14	80	10;22[a]	High	0.017(0.009±0.006)	0.120(0.057±0.042)	Pass
Fistula 4	192×256	14	80	10	High	0.003(0.002±0.001)	0.019(0.010±0.006)	Fail
Graft 1	192×256	14	80	10;10[b]	Low	0.000(0.000±0.000)	0.002(0.001±0.001)	Fail
Graft 1	192×256	14	80	10;8[b]	High	0.002(0.001±0.001)	0.010(0.005±0.003)	Fail
Graft 2	192×256	10	80	12;8[b]	Low	0.028(0.016±0.008)	0.190(0.102±0.056)	Fail
Graft 2	192×256	10	80	10	High	0.000(0.000±0.000)	0.001(0.000±0.000)	Pass
Graft 3	192×256	14	80	11;14[b]	Medium	0.009(0.006±0.002)	0.052(0.035±0.014)	Fail
Graft 3	192×256	14	80	12	High	0.001(0.001±0.000)	0.006(0.003±0.002)	Fail

[a] The second value gives the width of the cephalic vein

[b] The second value gives the width of the AV graft

5.2.5 3D Branch Cut Method

The 3D branch cut method was applied to the 4D PC-MRA from the volunteer and patient data. As was discussed previously in Section 3.4, some of the patient datasets were excluded due to their respective low v_{enc} settings causing severe corruption in the phase images.

The results are shown below in Table 5.9. The brachial artery was within the field of view for all the volunteers however the cephalic vein was not. All important vessels for haemodialysis, the brachial artery and either the cephalic vein or the e-PTFE graft, as well as the anastomosis were in the field of view for the patients' acquisitions. The Image Quality field was closely related to the number of residues and PSLs in the image. The L_p-norm evaluation shows the maximum, mean, and standard deviation of the L_p

values from all the frames of the dataset. The 4D PC-MRA data has velocity encoding in three spatial directions meaning there were three sets of 4D volumes. The number of residues, PSLs, and L_p-norm scores have three values each which represent the three encoding directions.

Table 5.9: 3D branch cut phase unwrapping applied to 4D PC-MRA

Patient ID	Image Size	№ of Slices	№ of Images	v_{enc} [cm/s]	ROI Width [px]	Image Quality	№ of Residues	№ of PSLs	L_1 Max (mean±sd)	L_2 Max (mean±sd)	Pass/Fail
Volunteer 1	128×96	20	20	60	6	High	280/634/477	139/163/121	0.034/0.000/0.000(0.002/0.000/0.000±0.008/0.000/0.000)	0.225/0.000/0.001(0.012/0.000/0.000±0.050/0.000/0.000)	Pass
Volunteer 2	128×96	20	14	20	5	Low	146475/143301/148915	29646/28982/29820	0.206/0.225/0.175(0.015/0.048/0.017±0.055/0.095/0.047)	1.294/1.412/1.100(0.097/0.303/0.104±0.345/0.594/0.294)	Fail
Volunteer 2	128×96	20	14	90	5	High	263/272/238	74/83/69	0.028/0.000/0.029(0.004/0.000/0.005±0.009/0.000/0.011)	0.177/0.000/0.188(0.027/0.438/0.035±0.060/0.000/0.070)	Pass
Volunteer 3	128×80	16	28	20	6	Medium	28322/21549/6307	3611/3040/1410	0.330/0.005/0.173(0.044/0.001/0.029±0.090/0.001/0.057)	2.540/0.034/1.699(0.339/0.004/0.281±0.704/0.007/0.572)	Fail
Volunteer 3	128×80	16	28	90	6	High	124/128/151	44/57/70	0.000/0.000/0.000(0.000/0.000/0.000±0.000/0.000/0.000)	0.000/0.000/0.000(0.000/0.000/0.000±0.000/0.000/0.000)	Pass
Volunteer 4	96×128	24	16	80	6	High	238/255/201	62/72/51	0.000/0.000/0.000(0.000/0.000/0.000±0.000/0.000/0.000)	0.000/0.000/0.000(0.000/0.000/0.000±0.000/0.000/0.000)	Pass
Volunteer 4	128×80	16	26	90	6	High	44/39/54	21/21/24	0.000/0.000/0.000(0.000/0.000/0.000±0.000/0.000/0.000)	0.000/0.000/0.000(0.000/0.000/0.000±0.000/0.000/0.000)	Pass
Volunteer 5	128×80	16	28	20	5	Low	131004/41516/71420	20216/9423/14444	0.133/0.004/0.006(0.019/0.002/0.004±0.032/0.001/0.001)	0.837/0.024/0.035(0.117/0.014/0.022±0.202/0.005/0.006)	Fail
Fistula 1	128×80	30	28	90	7;14[a]	Medium	23154/22879/30648	6036/6195/7493	0.001/0.0128/0.001(0.000/0.009/0.001±0.000/0.031/0.000)	0.007/0.941/0.005(0.003/0.064/0.003±.001/0.226/0.001)	Pass
Fistula 3	128×80	30	28	90	7;10[a]	Medium	36554/41359/58592	5308/9580/11572	0.005/0.105/0.002(0.003/0.008/0.001±0.001/0.027/0.000)	0.033/0.661/0.010(0.020/0.050/0.005±0.005/0.172/0.002)	Fail
Fistula 4	128×80	40	28	90	8;10[a]	Low	190848/236155/228578	37955/48094/45569	0.014/0.006/0.007(0.009/0.005/0.005±0.002/0.001/0.001)	0.088/0.038/0.045(0.050/0.028/0.034±0.015/0.003/0.004)	Pass
Fistula 5	128×80	32	12	250	8;10[a]	High	0/57/25	0/18/15	0.000/0.000/0.000(0.000/0.000/0.000±0.000/0.000/0.000)	0.000/0.000/0.000(0.000/0.000/0.000±0.000/0.000/0.000)	Pass
Graft 2	128×80	24	28	90	8;8[b]	Medium	24653/6550/3436	2644/1057/754	0.023/0.004/0.006(0.011/0.003/0.006±0.004/0.001/0.000)	0.154/0.029/0.041(0.056/0.018/0.035±0.030/0.004/0.003)	Pass

[a] The second value gives the width of the cephalic vein

[b] The second value gives the width of the AV graft

Chapter 6

Discussion

6.1 Validation

The algorithms were tested to determine at what point they would begin to fail. This was useful for determining which algorithms were susceptible to noise and low temporal or spatial sampling. Of the three algorithms used to unwrap the simulated 2D data, the temporal unwrapping technique was the least effective. It was susceptible to both noise and low temporal resolution which was expected. Unlike the other unwrapping techniques that were tested, temporal unwrapping is unable to avoid residues or areas with high noise; so any noise along the temporal axis will corrupt that line of pixels. Similarly if the sampling rate is too low the temporal resolution will not be high enough to accurately capture the dynamics of the blood flow curve. The chosen sampling rate must ensure that the absolute change in velocity between pixels in consecutive frames never exceeds v_{enc}.

$$\frac{|v(t + \text{TR}) - v(t)|}{\text{TR}} \leq \frac{v_{\text{enc}}}{\text{TR}}$$

In Figure 6.1 a noiseless, simulated blood flow curve is given with two different sampling frequencies. To illustrate how the choice of v_{enc} and TR affect the temporal algorithm's ability to unwrap the sampled blood flow curve, consider Figure 6.1a. The long TR means that the v_{enc} must be greater than approximately 30 cm/s so that no information is lost when the curve is wrapped. This is illustrated in Figure 6.1b, where the samples taken at t and t+TR are clearly wrapped. However, due to the low sampling rate, the

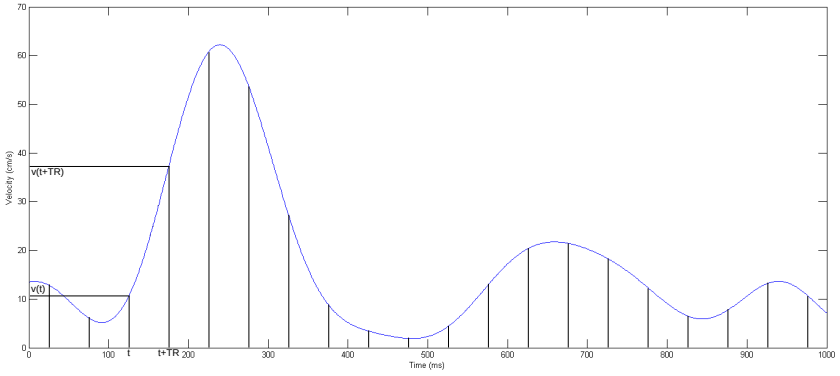
absolute difference between their two velocities is less than v_{enc} and the wrap cannot be detected. For Figure 6.1c the decreased TR means that the v_{enc} parameter can be lowered without losing information which is illustrated by Figure 6.1d; the samples marked t and t+TR are wrapped and because the absolute difference between their velocities is higher than the v_{enc} parameter, the wrap is detected and can be removed.

The most robust techniques for unwrapping the simulated 2D datasets were the Goldstein-Zebker-Werner branch cut method and the PDV Quality Guided Phase Unwrapping algorithm.

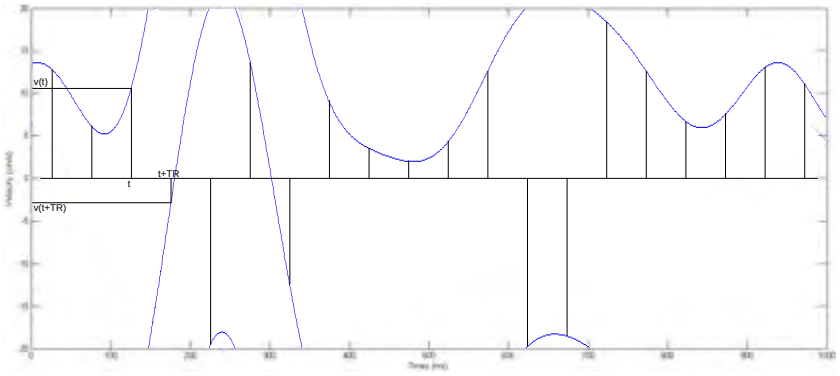
The Goldstein branch cut method was able to handle noise more effectively than the temporal phase unwrapping algorithm as it only began to fail when the noise variance was set to 0.05. This noise variance value means that approximately 15% of the image is residues which is similar to the noise density of 1 residues per 10 pixels at which other unwrapping techniques begin to fail as reported by Cusack et al. (1995). The size of the test images were kept at a constant 50×50 pixels while the diameter of the ROI was varied which means the effective spatial resolution of the ROI could be altered. Due to the laminar flow of blood, the blood flow velocities do vary from little to no flow on the perimeter of the vessel to peak flow at the centre. This means that the sampling rate for the spatial axes must ensure that the absolute change in velocity between adjacent pixels do not exceed v_{enc}, similar to the temporal sampling rate. The Goldstein branch cut method, however, was less susceptible to issues caused by low resolution provided the v_{enc} parameter was set at an appropriate value.

The PDV QGPU algorithm performed equally as effectively as the Goldstein branch cut method.

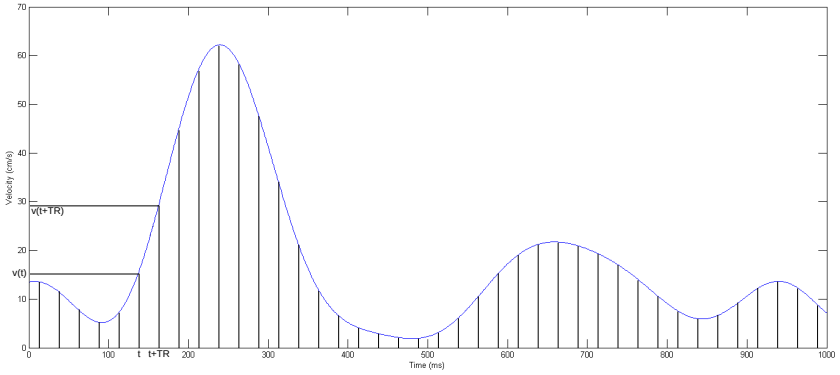
The 3D branch cut method was used to unwrap a series of simulated 3D volumes. As with the simulated 2D data, the noise variance could be changed to vary the noise density and in doing so vary the number of phase residues present in the volume. The algorithm performed well in the presence of noise, even being able to unwrap volumes of size 30×30×30 and larger when the noise variance was set to 0.05, causing 15% of the pixels to be corrupted. The volume dimensions and the v_{enc} scaling factor work in concert to adjust the distance between wraps, in essence adjusting the spatial resolution. A large scaling factor and small dimensions means that the distance between wraps will be very small, potentially leading to a situation where information is lost between adjacent



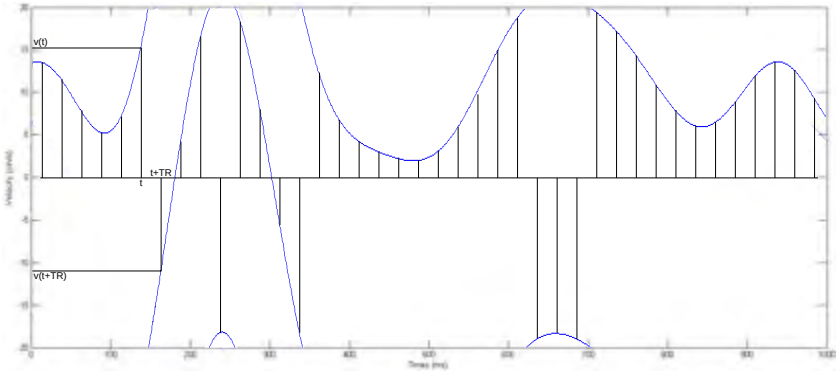
(a) Flow curve sampled with a TR of 50 ms



(b) Wrapped flow curve with a v_{enc} of 20 cm/s



(c) Flow curve sampled with a TR of 25 ms



(d) Wrapped flow curve with a v_{enc} of 20 cm/s

Figure 6.1: A simulated blood flow curve with a length of 1 000 ms

pixels. This, coupled with the high noise density, is the reason the volume in Figure 5.4b could not be unwrapped successfully.

6.2 PC-MRA Data

Having validated the unwrapping algorithms on simulated datasets, the algorithms were finally applied to the in vivo subject data. The temporal phase unwrapping algorithm, Goldstein branch cut method, and PDV QGPU were applied to the 2D PC-MRA datasets. The temporal phase unwrapping algorithm was only able to be applied to the volunteer datasets as the majority of patient datasets contained phase wrapping in every frame and as such there was no reference frame from which to begin the unwrapping algorithm. The algorithm was able to unwrap 2 of 9 datasets successfully; the failures were due primarily to a combination of low spatial resolution and low v_{enc} settings.
The Goldstein branch cut method and the QGPU algorithm were applied to all subject data. The Goldstein method was able to unwrap 3 of 9 volunteer datasets and 5 of 14 patient datasets while the QGPU algorithm successfully unwrapped 3 of 9 volunteer datasets and 4 of 14 patient datasets. The low v_{enc} settings of the volunteer datasets is the primary reason for those failed unwrapping attempts. The patient data was somewhat more difficult to accurately judge as successfully or unsuccessfully unwrapped due to the complex nature of the blood flow throughout the haemodialysis connections. Often non-laminar flow was encountered, especially in the patients' anastomoses which made determining the actual direction of blood flow in a given frame difficult.
Due to the problem encountered with the patient datasets when using the temporal phase unwrapping technique coupled with the failures of the spatial unwrapping techniques, a composite technique was implemented. The Goldstein branch cut method was used in conjunction with the temporal phase unwrapping to provide a robust unwrapping technique. Although the patient datasets had a similar temporal resolution to the volunteer datasets, which previously caused the unwrapping procedure to fail, the blood flow in haemodialysis access vessels is less pulsatile than normal blood flow. This means that the lower temporal resolution would not necessarily be detrimental to the temporal unwrapping technique. The first frame of each dataset was unwrapped using the Goldstein branch cut method, this frame is then used as a reference frame with which to temporally unwrap the remaining frames in each dataset. This method proved to be

effective as it was able to unwrap 8 of 14 patient datasets successfully.

The 4D PC-MRA subject data was unwrapped using the 3D branch cut method. The algorithm was able to unwrap 5 of 8 volunteer datasets and 4 of 5 patient datasets. The most commonly encountered problem that would lead to an unsuccessful unwrapping attempt was poor image quality. In order to keep the 4D PC-MRA acquisition times, which in general are very long, to a reasonable time frame, it was necessary to reduce the spatial resolution of the scans. However, due to the robust design of the 3D branch cut method, it was able to unwrap the majority of 4D datasets.

Chapter 7

Conclusions and Future Work

Phase wrapping is a common problem which is encountered in several different kinds of MR imaging; especially, for this work, with velocity encoded PC-MRA. The choice of v_{enc} is of utmost importance during the acquisition of subject data as a value that is too low can lead to heavy phase wrapping, potentially corrupting the image beyond repair, while a value that is too high can lead to low SNR leading to a situation where signal from flow is indistinguishable from the background signal. As such setting the v_{enc} parameter to be slightly lower than the expected highest velocity can be beneficial as the SNR is increased and any phase wrapping can be removed during pre-processing. However, to choose an appropriate value for the v_{enc} parameter can be difficult especially when faced with complex anatomical vessels, as is the case with haemodialysis patients. Blood flow velocity, and vessel diameter and shape can vary widely between subjects and even within a single subject. For this reason it is useful to perform several, fast "scouting" acquisitions throughout the patient's anatomy in order to determine the ideal scanning parameters for that patient and thus avoid most of the problems associated with phase wrapping as well as low spatial and temporal resolution.

Several methods of phase unwrapping have been investigated and tested using both simulated data, for the purposes of validation, and subject data. The algorithms chosen for testing were based on recommendations from Ghiglia & Pritt (1998) and Salfity et al. (2004).

The temporal phase unwrapping algorithm performed the most poorly of the four methods for unwrapping 2D datasets, this was due in part to a potential flaw in the algorithm

that requires a known reference point which can be problematic if there is phase wrapping throughout the dataset. The algorithm also requires sufficient temporal resolution to capture the dynamics of the flow signal otherwise information is lost beyond retrieval. The Goldstein-Zebker-Werner branch cut algorithm and the PDV QGPU algorithm were both effective methods of unwrapping 2D datasets. The algorithms were largely immune to noise but would require sufficient spatial resolution in the ROI so that information from the signal is not lost in the presence of phase wrapping.

The most effective technique for unwrapping the 2D datasets was the spatio-temporal phase unwrapping algorithm, a combination of the temporal phase unwrapping and the Goldstein branch cut algorithms. Although improved, it is not immune to low temporal or spatial resolution, however, due to the unique circumstances related to the creation of haemodialysis access it was especially effective in unwrapping the patient PC-MRA datasets.

Finally, the 3D branch cut algorithm was an effective method of unwrapping 3D datasets. The algorithm performed very well even in the presence of high noise and was able to unwrap the majority of 4D PC-MRA datasets.

There are several other methods for unwrapping phase images that were considered but not implemented due to time constraints that may have potentially provided improved results including region merging (Jenkinson 2003, Ongori et al. 2008, Zhou et al. 2009) and L_p-norm based methods (Ghiglia & Pritt 1998). In a recent article by Untenberger et al. (2015), an extension to the spatio-temporal phase unwrapping algorithm was proposed that would improve the robustness of the algorithm used in this project.

A further extension to this project could be to develop a program to visualise the blood flow in the haemodialysis patients, this would be a useful development for determining the success of an unwrapping operation more quickly as incorrectly unwrapped areas would be immediately visible without having to go through an entire set of images. Some techniques that discuss the visualisation of blood flow can be found in Buonocore (1998), Wigström et al. (1999). Other developments that could be added are the ability to determine wall shear stress (Oyre et al. 1998) in the haemodialysis access vessels, especially in the anastomosis, and non-invasively determine the blood pressure (Bock et al. 2011) in the access vessels.

Appendix A

Residue Lookup Table

Table A.1: Lookup table of possible residues and their proceeding connections

Current Residue		Joining Residues		Offset		
Type	Charge	Type	Charge	x	y	z
x	1	x	1	+1	0	0
		y	1	0	+1	0
		y	-1	0	0	0
		z	1	0	0	+1
		z	-1	0	0	0
	-1	x	-1	-1	0	0
		y	1	-1	+1	0
		y	-1	-1	0	0
		z	1	-1	0	+1
		z	-1	-1	0	0
y	1	x	1	+1	0	0
		x	-1	0	0	0
		y	1	0	+1	0
		z	1	0	0	+1
		z	-1	0	0	0
	-1	x	1	+1	-1	0
		x	-1	0	-1	0
		y	-1	0	-1	0
		z	1	0	-1	+1
		z	-1	0	-1	0
z	1	x	1	+1	0	0
		x	-1	0	0	0
		y	1	0	+1	0
		y	-1	0	0	0
		z	1	0	0	+1
	-1	x	1	+1	0	-1
		x	-1	0	0	-1
		y	1	0	+1	-1
		y	-1	0	0	-1
		z	-1	0	0	-1

Table A.2: Lookup table of possible residues and their preceding connections

Current Residue		Joining Residues		Offset		
Type	Charge	Type	Charge	x	y	z
x	1	x	1	-1	0	0
		y	1	-1	0	0
		y	-1	-1	+1	0
		z	1	-1	0	0
		z	-1	-1	0	+1
	-1	x	-1	+1	0	0
		y	1	0	0	0
		y	-1	0	+1	0
		z	1	0	0	0
		z	-1	0	0	+1
y	1	x	1	0	-1	0
		x	-1	+1	-1	0
		y	1	0	-1	0
		z	1	0	-1	0
		z	-1	0	-1	+1
	-1	x	1	0	0	0
		x	-1	+1	0	0
		y	-1	0	+1	0
		z	1	0	0	0
		z	-1	0	0	+1
z	1	x	1	0	0	-1
		x	-1	+1	0	-1
		y	1	0	0	-1
		y	-1	0	+1	-1
		z	1	0	0	-1
	-1	x	1	0	0	0
		x	-1	+1	0	0
		y	1	0	0	0
		y	-1	0	+1	0
		z	-1	0	0	+1

Bibliography

Akoh, J. A. (2009), 'Prosthetic arteriovenous grafts for hemodialysis', *Journal of Vascular Access* **10**(12), 137–147.

Al-Jaishi, A. a., Oliver, M. J., Thomas, S. M., Lok, C. E., Zhang, J. C., Garg, A. X., Kosa, S. D., Quinn, R. R. & Moist, L. M. (2014), 'Patency rates of the arteriovenous fistula for hemodialysis: A systematic review and meta-analysis', *American Journal of Kidney Diseases* **63**(3), 464–478.

Albayrak, R., Yuksel, S., Colbay, M., Degirmenci, B., Acarturk, G., Haktanir, A. & Karaman, O. (2007), 'Hemodynamic changes in the cephalic vein of patients with hemodialysis arteriovenous fistula', *Journal of Clinical Ultrasound* **35**(3), 133–137.

Ascher, E. & Hingorani, A. (2004), 'The Dialysis Outcome and Quality Initiative (DOQI) Recommendations', *Seminars in Vascular Surgery* **17**(1), 3–9.

Ates, A., Ozyazicioglu, A., Yekeler, I., Ceviz, M., Erkut, B., Karapolat, S., Koçogullari, C. U. & Kocak, H. (2006), 'Primary and secondary patency rates and complications of upper extremity arteriovenous fistulae created for hemodialysis.', *The Tohoku Journal of Experimental Medicine* **210**, 91–97.

Bernstein, M. A., King, K. E. & Zhou, X. J. (2004), *Handbook of MRI Pulse Sequences*, Elsevier Academic Press, Burlington, MA, USA.

Bieri, O. & Scheffler, K. (2013), 'Fundamentals of balanced steady state free precession MRI', *Journal of Magnetic Resonance Imaging* **38**, 2–11.

Bock, J., Frydrychowicz, A., Lorenz, R., Hirtler, D., Barker, A. J., Johnson, K. M., Arnold, R., Burkhardt, H., Hennig, J. & Markl, M. (2011), 'In vivo noninvasive 4D pressure difference mapping in the human aorta: phantom comparison and application

in healthy volunteers and patients.', *Magnetic Resonance in Medicine* **66**(4), 1079–1088.

Bock, J., Kreher, B. W., Hennig, J. & Markl, M. (2007), Optimized pre-processing of time-resolved 2D and 3D phase contrast MRI data, *in* '15th Annual Meeting of ISMRM', ISMRM, Berlin, DEU, p. 3138.

Brown, R. W., Cheng, Y.-C. N., Haacke, E. M., Thompson, M. R. & Venkatesan, R. (2014), *Magnetic Resonance Imaging Physical Principles and Sequence Design*, second edn, John Wiley & Sons, Hoboken, NJ, USA.

Buonocore, M. H. (1998), 'Visualizing Blood Flow Patterns Using Streamlines, Arrows, and Particle Paths', *Magnetic Resonance in Medicine* **40**(2), 210–226.

Bushberg, J. T., Seibert, J. A., Leidholdt Jr, E. M. & Boone, J. M. (2001), *The Essential Physics of Medical Imaging*, 2 edn, Lippincott Williams & Wilkins, Philadelphia, PA, USA.

Clement, F. M., Klarenbach, S., Tonelli, M., Johnson, J. A. & Manns, B. J. (2009), 'The Impact of Selecting a High Hemoglobin Target Level on Health-Related Quality of Life for Patients With Chronic Kidney Disease: A Systematic Review and Meta-analysis', *Archives of Internal Medicine* **169**(12), 1104–1112.

Cusack, R., Huntley, J. M. & Goldrein, H. T. (1995), 'Improved noise-immune phase-unwrapping algorithm.', *Applied Optics* **34**(5), 781–789.

Cusack, R. & Papadakis, N. (2002), 'New Robust 3-D Phase Unwrapping Algorithms: Application to Magnetic Field Mapping and Undistorting Echoplanar Images', *NeuroImage* **16**(3), 754–764.

Drawz, P., Rahman, M., Laine, C., Goldmann, D. R. & Sox, H. C. (2009), 'Chronic kidney disease', *Annals of Internal Medicine* **150**(3), 714–721.

Field, M., McNamara, K., Bailey, G., Jaipersad, A., Morgan, R. H. & Pherwani, A. D. (2008), 'Primary patency rates of AV fistulae and the effect of patient variables', *Journal of Vascular Access* **9**(1), 45–50.

Fox, S. I. (2011), *Human Physiology*, 12 edn, McGraw-Hill, New York, NY, USA.

Ghiglia, D. C. & Pritt, M. D. (1998), *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*, John Wiley & Sons, New York, NY, USA.

Gibbons, C. P. (2006), 'Primary Vascular Access', *European Journal of Vascular and Endovascular Surgery* **31**(5), 523–529.

Gibson, K. D., Gillen, D. L., Caps, M. T., Kohler, T. R., Sherrard, D. J. & Stehman-Breen, C. O. (2001), 'Vascular access survival and incidence of revisions: a comparison of prosthetic grafts, simple autogenous fistulas, and venous transposition fistulas from the United States Renal Data System Dialysis Morbidity and Mortality Study.', *Journal of Vascular Surgery* **34**(4), 694–700.

Gilpin, V. & Nichols, W. (2010), 'Vascular access for hemodialysis: Thrills and thrombosis', *Journal of Vascular Nursing* **28**(2), 78–83.

Goldstein, R. M., Zebker, H. a. & Werner, C. L. (1988), 'Satellite radar interferometry: Two-dimensional phase unwrapping', *Radio Science* **23**(4), 713–720.

Herrick, A. L. & Hutchinson, C. (2004), 'Vascular imaging.', *Best Practice & Research. Clinical Rheumatology* **18**(6), 957–979.

Huang, T.-C., Chang, C.-K., Liao, C.-H. & Ho, Y.-J. (2013), 'Quantification of blood flow in internal cerebral artery by optical flow method on digital subtraction angiography in comparison with time-of-flight magnetic resonance angiography.', *PLOS ONE* **8**(1), e54678.

Huijbregts, H. J. T., Bots, M. L., Wittens, C. H. A., Schrama, Y. C., Moll, F. L. & Blankestijn, P. J. (2008), 'Hemodialysis arteriovenous fistula patency revisited: Results of a prospective, multicenter initiative', *Clinical Journal of the American Society of Nephrology* **3**(10), 714–719.

Hung, Y.-N., Ko, P.-J., Ng, Y.-Y. & Wu, S.-C. (2010), 'The Longevity of Arteriovenous Graft for Hemodialysis Patients – Externally Supported or Nonsupported', *Clinical Journal of the American Society of Nephrology* **5**(6), 1029–1035.

Huntley, J. M. (2001), 'Three-dimensional noise-immune phase unwrapping algorithm', *Applied Optics* **40**(23), 3901–3908.

Huntley, J. M. & Saldner, H. O. (1993), 'Temporal phase-unwrapping algorithm for automated interferogram analysis', *Applied Optics* **32**(17), 3047–3052.

Itoh, K. (1982), 'Analysis of the phase unwrapping algorithm.', *Applied Optics* **21**(14), 2470.

Jara, H. & Barish, M. A. (1999), 'Black-blood MR angiography. Techniques, and clinical applications', *Magnetic Resonance Imaging Clinics of North America* **7**, 303–317.

Jenkinson, M. (2003), 'Fast, automated, N-dimensional phase-unwrapping algorithm', *Magnetic Resonance in Medicine* **49**(1), 193–197.

Jindal, K., Chan, C. T., Deziel, C., Hirsch, D., Soroka, S. D., Tonelli, M. & Culleton, B. F. (2006), 'Hemodialysis clinical practice guidelines for the Canadian Society of Nephrology.', *Journal of the American Society of Nephrology* **17**, S1–S27.

KDOQI (2006), 'Vascular access', *American Journal of Kidney Diseases* **48 Suppl 1**, S227–S409.

Kim, S.-e. & Parker, D. L. (2012), Time-of-Flight Angiography, *in* J. C. Carr & T. J. Carroll, eds, 'Magnetic Resonance Angiography: Principles and Applications', Springer New York, New York, NY, USA, chapter 2, pp. 39–51.

Lankhaar, J. W., Hofman, M. B. M., Marcus, J. T., Zwanenburg, J. J. M., Faes, T. J. C. & Vonk-Noordegraaf, A. (2005), 'Correction of phase offset errors in main pulmonary artery flow quantification', *Journal of Magnetic Resonance Imaging* **22**(1), 73–79.

Levitt, M. H. (2013), Ensembles of Spin-1/2, *in* 'Spin Dynamics: Basics of Nuclear Magnetic Resonance', 2 edn, John Wiley & Sons, Chichester, GBR, chapter 11, pp. 259–294.

Lorenz, R., Bock, J., Snyder, J., Korvink, J. G., Jung, B. a. & Markl, M. (2014), 'Influence of eddy current, Maxwell and gradient field corrections on 3D flow visualization of 3D CINE PC-MRI data', *Magnetic Resonance in Medicine* **72**(1), 33–40.

Lotz, J., Meier, C., Leppert, A. & Galanski, M. (2002), 'Cardiovascular flow measurement with phase-contrast MR imaging: basic facts and implementation.', *Radiographics* **22**(3), 651–671.

Lowth, M. (2013), 'Chronic kidney disease – an update', *Practice Nurse* **43**(1), 34–39.

Lu, Y., Wang, X. & Zhang, X. (2007), 'Weighted least-squares phase unwrapping algorithm based on derivative variance correlation map', *Optik* **118**(2), 62–66.

Lu, Y., Zhao, W., Zhang, X., Xu, W. & Xu, G. (2012), 'Weighted-phase-gradient-based quality maps for two-dimensional quality-guided phase unwrapping', *Optics and Lasers in Engineering* **50**(10), 1397–404.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S0143816612001510*

Mansilla, A. V., Toombs, B. D., Vaughn, W. K. & Zeledon, J. I. (2001), 'Patency and Life-Spans of Failing Hemodialysis Grafts', *Texas Heart Institute Journal* **28**(4), 249–253.

Markl, M. (2005), 'Velocity Encoding and Flow Imaging'.
URL: *http://ee-classes.usc.edu/ee591/library/Markl-FlowImaging.pdf*

Markl, M., Frydrychowicz, A., Kozerke, S., Hope, M. & Wieben, O. (2012), '4D Flow MRI', *Journal of Magnetic Resonance Imaging* **36**(5), 1015–1036.

Markl, M., Kilner, P. J. & Ebbers, T. (2011), 'Comprehensive 4D velocity mapping of the heart and great vessels by cardiovascular magnetic resonance', *Journal of Cardiovascular Magnetic Resonance* **13**(1), 7.

Markl, M. & Leupold, J. (2012), 'Gradient echo imaging', *Journal of Magnetic Resonance Imaging* **35**, 1274–1289.

Marklund, O., Huntley, J. M. & Cusack, R. (2005), Robust Unwrapping Algorithm for 3-D Phase Volumes of Arbitrary Shape Containing Knotted Phase Singularity Loops, Technical report, LuleåUniversity of Technology, Luleå, SWE.

Matsuura, J. H., Rosenthal, D., Clark, M., Shuler, F. W., Kirby, L., Shotwell, M., Purvis, J. & Pallos, L. L. (1998), 'Transposed Basilic Vein Versus Polytetrafluorethylene for Brachial-Axillary Arteriovenous Fistulas', *American Journal of Surgery* **176**(2), 219–221.

Meyers, A. M. (2015), 'Chronic kidney disease', *South African Medical Journal* **105**(3), 232.
URL: *http://hmpg.co.za/index.php/samj/article/view/2714*

Murphy, S. L., Xu, J. & Kochanek, K. D. (2013), 'Deaths: Final Data for 2010', *National Vital Statistics Reports* **61**(4), 1–167.

Naicker, S. (2003), 'End-stage renal disease in sub-Saharan and South Africa', *Kidney International* **63**(83), S119–S122.

Ongori, J. N., Meintjes, E. M. & Spottiswoode, B. S. (2008), 3D Phase Unwrapping of DENSE MRI Images Using Region Merging, *in* F. Nicolls, ed., 'Pattern Recognition Association of South Africa', Cape Town, ZAF, pp. 119–124.

Ostermann, M. (2011), Choice of Renal Replacement Therapy and Role of Haemodialysis in the Intensive Care Unit, *in* M. G. Penido, ed., 'Hemodialysis - different aspects', InTech, Rijeka, HRV, chapter 15, pp. 253–262.

Oyre, S., Paaske, W. P., Ringgaard, S., Kozerke, S., Erlandsen, M., Boesiger, P. & Pedersen, E. M. (1998), 'Automatic Accurate Non-invasive Quantitation of Blood Flow, Cross-sectional Vessel Area, and Wall Shear Stress by Modelling of Magnetic Resonance Velocity Data', *European Journal of Vascular and Endovascular Surgery* **16**(6), 517–524.

Ozcan, H., Öztekin, P. S., Zergeroğlu, A. M., Ersöz, G., Fiçicilar, H. & Üstüner, E. (2006), 'Doppler ultrasound evaluation of the structural and hemodynamic changes in the brachial artery following two different exercise protocols', *Diagnostic and Interventional Radiology* **12**(2), 80–84.

Patel, P. P., Altieri, M., Jindal, T. R., Guy, S. R., Falta, E. M., Elster, E. A., Hurst, F. P., Sidawy, A. N. & Jindal, R. M. (2011), Current Status of Synthetic and Biological Grafts for Hemodialysis, *in* M. G. Penido, ed., 'Hemodialysis - different aspects', InTech, Rijeka, HRV, chapter 17, pp. 285–312.

Planken, R. N., Tordoir, J. H. M., Duijm, L. E. M., de Haan, M. W. & Leiner, T. (2007), 'Current techniques for assessment of upper extremity vasculature prior to hemodialysis vascular access creation', *European Radiology* **17**(11), 3001–3011.

Roditi, G. (2002), 'Contrast-enhanced magnetic resonance angiography.', *The British Journal of Surgery* **89**, 817–820.

Salfity, M. F., Huntley, J. M., Graves, M. J., Marklund, O., Cusack, R. & Beauregard, D. A. (2004), 3-D and 4-D phase unwrapping methods applied to phase contrast magnetic resonance velocity imaging, *in* '12th International Conference on Experimental Mechanics', Bari, ITA, pp. 3–8.

Shoji, T., Abe, T., Matsuo, H., Egusa, G., Yamasaki, Y., Kashihara, N., Shirai, K. & Kashiwagi, A. (2012), 'Chronic Kidney Disease, Dyslipidemia, and Atherosclerosis', *Journal of Atherosclerosis and Thrombosis* **19**(4), 299–315.

South African Renal Society (2006), South African Renal Society Recommendations for Early Detection and Management of Chronic Kidney Disease, Technical report, South African Renal Society, ZAF.
URL: *http://www.inpracticeafrica.com/ /media/Guidelines/SA_Ren_Disease.pdf*

Spottiswoode, B. S. (2006), Towards automating cine DENSE MRI image analysis : segmentation , tissue tracking and strain computation, Ph.d, University of Cape Town.
URL: *http://uctscholar.uct.ac.za/PDF/112513_Spottiswoode_B.S..pdf*

Stalder, A. F., Russe, M. F., Frydrychowicz, A., Bock, J., Hennig, J. & Markl, M. (2008), 'Quantitative 2D and 3D phase contrast MRI: optimized analysis of blood flow and vessel wall parameters.', *Magnetic Resonance in Medicine* **60**(5), 1218–1231.

Statistics South Africa (2013), Mortality and causes of death in South Africa, 2010: Findings from death notification, Technical Report November, Statistics South Africa, Pretoria, ZAF.
URL: *http://www.statssa.gov.za/publications/p03093/p030932010.pdf*

Statistics South Africa (2014), Mortality and causes of death in South Africa, 2013: Findings from death notification, Technical Report July, Statistics South Africa, Pretoria, ZAF.
URL: *http://www.statssa.gov.za/publications/P03093/P030932013.pdf*

Suprijanto, Vrooman, H. & Vossepoel, A. (2001), Quality maps to support phase unwrapping, *in* R. Lagendijk, J. Heijnsdijk, A. Pimentel & M. Wilkinson, eds, 'Proceedings of ASCI 2001, 7th Annual Conference of the Advanced School for Computing and Imaging', Heijen, NLD, pp. 220–226.

Tsimihodimos, V., Mitrogianni, Z. & Elisaf, M. (2011), 'Dyslipidemia Associated with Chronic Kidney Disease', *The Open Cardiovascular Medicine Journal* **5**, 41–48.

Untenberger, M., Hüllebrand, M., Tautz, L., Joseph, A. A., Voit, D., Merboldt, K. D. & Frahm, J. (2015), 'Spatiotemporal phase unwrapping for real-time phase-contrast flow MRI', *Magnetic Resonance in Medicine* **74**(4), 964–970.
URL: *http://doi.wiley.com/10.1002/mrm.25471 http://www.ncbi.nlm.nih.gov/pubmed/25302683*

Uppman, M. (2009), Constrained Periodic Particle Tracking Using MRI, Masters, Lund University.

U.S. Renal Data System (2013*a*), 'Volume 1: Atlas of Chronic Kidney Disease in the United States', *American Journal of Kidney Diseases* **61**(1 Suppl 1), e1–e148.

U.S. Renal Data System (2013*b*), 'Volume 2: Atlas of End-Stage Renal Disease in the United States', *American Journal of Kidney Diseases* **61**(1 Suppl 1), e149–e480.

Wigström, L., Ebbers, T., Fyrenius, A., Karlsson, M., Engvall, J., Wranne, B. & Bolger, A. F. (1999), 'Particle Trace Visualization of Intracardiac Flow Using Time-Resolved 3D Phase Contrast MRI', *Magnetic Resonance in Medicine* **41**(4), 793–799.

Wigström, L., Sjöqvist, L. & Wranne, B. (1996), 'Temporally resolved 3D phase contrast imaging', *Magnetic Resonance in Medicine* **36**(9), 800–803.

Ying, L. (2006), Phase Unwrapping, *in* M. Akay, ed., 'Wiley Encyclopedia of Biomedical Engineering', 1st edn, John Wiley & Sons, Hoboken, NJ, USA.

Zhang, Y., Wang, S., Ji, G. & Dong, Z. (2014), 'An Improved Quality Guided Phase Unwrapping Method and Its Applications to MRI', *Progress in Electromagnetics Research* **145**(April), 273–286.

Zhou, K., Zaitsev, M. & Bao, S. (2009), 'Reliable two-dimensional phase unwrapping method using region growing and local linear estimation.', *Magnetic Resonance in Medicine* **62**(4), 1085–1090.