A seismic study of the mid- and lower-crust beneath the sea of Bothnia: BABEL line 1

Daniel P. Graham

A Seismic Study of the Mid- and Lower-Crust Beneath the Sea of Bothnia: BABEL Line 1

Daniel P. Graham

Graduate Society

A thesis presented for the degree of Doctor of Philosophy

Department of Geological Sciences,
University of Durham.

December 1994

A Seismic Study of the Mid- and Lower-Crust Beneath the Sea of Bothnia: BABEL Line 1

Abstract

In the Autumn of 1989, Durham University took part in the BABEL Project, a collaboration of scientists from five nations recording wide angle and normal incidence seismic data in the Baltic Shield. Recording stations were set up along the Swedish coast of the Sea of Bothnia to record marine airgun shots at wide angle. Similar stations were operated by Finnish teams on the eastern coast, and by a German team on Åland. The data recorded are of high quality and high resolution in comparison to previous wide angle surveys in the region, with a shot spacing of 75 m.

A large proportion (around fifty percent) of the project involved developing software for processing this data.

The in-line data from Line 1, and also those recorded at two off-line stations, have been interpreted using Cerveny's Gaussian Beam forward modelling package BEAM87, the in-line model being further constrained by 2D gravity modelling. The resulting models are compared and contrasted with normal incidence data from the same line, other models derived from BABEL data in the Sea of Bothnia and older refraction lines in the vicinity.

The models show a highly complex crust whose thickness varies between 50 and 60 km along the profile. The seismic velocity is high, increasing from 5.85 km s^{-1} near the surface to 7.4 km s^{-1} at the base of the crust. Lateral velocity variations are seen in the mid-upper crust while discontinuous reflectors and diffracting bodies are seen at 30 km depth.

In the central/northern part of the line the crust thickens and there is a change in seismic velocity. Using other geophysical information from the region, two hypotheses are put forward for interpreting this part of the seismic model. The first is the presence of a large igneous intrusion, and the second is the existence of a shear zone or tectonic boundary cutting Line 1. Further work will be required to confirm either or both of these hypotheses.

Department of Geological Sciences,
University of Durham.

Declaration

The work contained in this thesis has not been previously submitted for any other degree or qualification and that unless otherwise stated is the author's own work.

Acknowledgments

I would like to thank my supervisor, Roger Long, for offering me this opportunity in the first place; Tricia Matthews for encouraging me along the way; Eric Lippman who was largely responsible for organising and carrying out the fieldwork; Dave Stevenson for his advice and assistance on the computing side, and the technical staff at Durham. I would also like to thank my wife, Anna, for being supportive and putting up with me during the writing of this thesis.

The BABEL Working Group consisted of:

Denmark: A. Berthelsen, H. Thybo (Institute of Geology, University Of Copenhagen); N. Balling, E. Normark (Department of Earth Sciences, Aarhus University); T. Dahl-Jensen (geological Survey of Greenland).

Finland: P. Heikkinen, K. Korhonen, U. Luosto (Institute of Seismology, Helsinki University); S. E. Hjelt, K. Komminaho, J. Yliniemi (Geophysics Observatory, Oulu University).

Germany: T. Dickman, E. R. Flueh (GEOMAR, Kiel); R. Meissner, P. Sadowiak, S. A. Thomas, Th. Wever (Institute for Geophysics, University of Kiel).

Sweden: C.-E. Lund, H. Palm, L. B. Pedersen, R. G. Roberts (Solid Earth Geophysics, Uppsala University); S. A. Elming (Tillampad Geofysik Lulea Technical University).

United Kingdom: R. W. Hobbs, S. L. Klemperer, D. H. Matthews, D. B. Snyder (BIRPS, Bullard Laboratories, Cambridge University); R. E. Long, D. P. Graham, P. A. Matthews (Department of Geological Sciences, University of Durham); D. J. Blundell, R. Scott-Robinson (Department of Geology, RHB New College).

The marine seismic work was carried out by the contractors PRAKLA-SEISMOS of Hannover using their vessel M/V MINTROP.

The BABEL Project was sponsored by:

NERC

The Commission of the European Community (JOULE Project)

BIRPS (British Institutions Reflection Profiling Syndicate)

Industrial Associates (Amerada-Hess Ltd, BP Exploration Co. Ltd, Britoil, Chevron UK Ltd, Conoco (UK) Ltd, Enterprise Oil plc, Mobil North Sea plc, Shell UK Exploration and Production)

Swedish Natural Science Research Council

The Academy of Finland

Deutsche Forschungsgemeinschaft

Danish Natural Science Research Council

Contents

List of Figures

List of Tables

1. Prologue

In September 1989, 2268 km of marine seismic lines were shot between the north east coast of Germany and the Bothnian Bay (Fig. 1.1). Over 40 000 airgun shots were recorded, both at normal incidence and at wide angle by 68 land based seismic stations spread all around the Baltic coast.

The original aims of the BABEL project (BalticAnd Bothnian Echoes from the Lithosphere) were twofold. Firstly the desire to compare the deep marine seismic data obtained around the British Isles and Western Europe with a similar dataset from a different crustal type. Secondly the need for a deep seismic reflection study of the transition from the Mesozoic and Paleozoic crust of Denmark and Germany to the Proterozoic crust of Scandinavia, and the Tornquist Zone, to compliment data from the EUGENO-S project of 1984.

The project quickly expanded, taking on more aims and objectives within the Baltic Sea and Gulf of Bothnia region. Some of these will be discussed further in Chapter 4, after an account of the geological background of the region and the results of previous geophysical investigations. Implicit in many of the aims was the necessity for simultaneous recording of off-line wide angle data to provide three dimensional velocity and reflection information. The experiment became a major international project involving the collaboration of 31 scientists from 12 institutions and 5 nations (Denmark, Finland, Germany, Sweden and the United Kingdom).

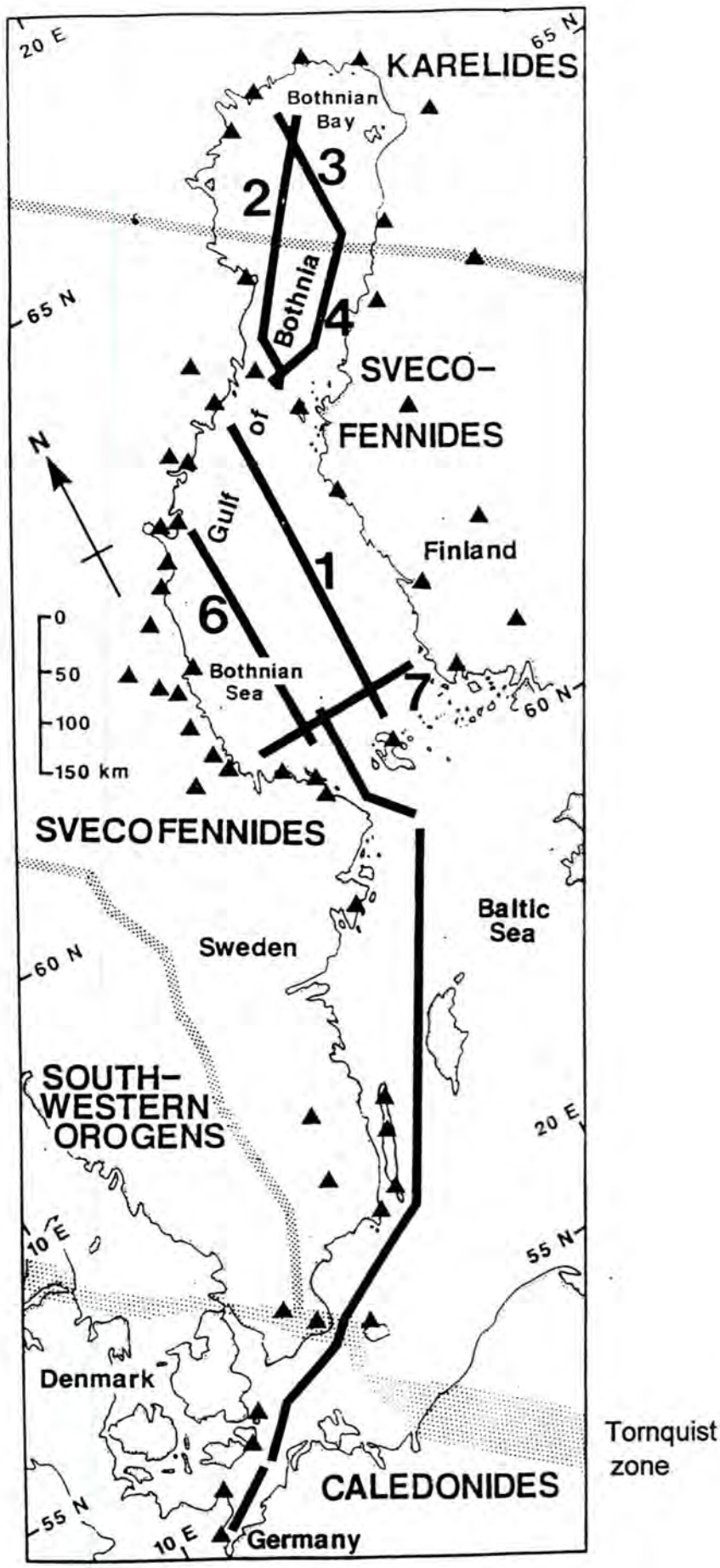


Fig. 1.1 Location map of normal incidence profiles (thick solid lines) recorded during Project BABEL and seismic stations (triangles) used to record the marine seismic source at wide angles. Stippled lines indicate major tectonic boundaries (BABEL Working Group, 1990).

2. Geological Background

2.1 Introduction

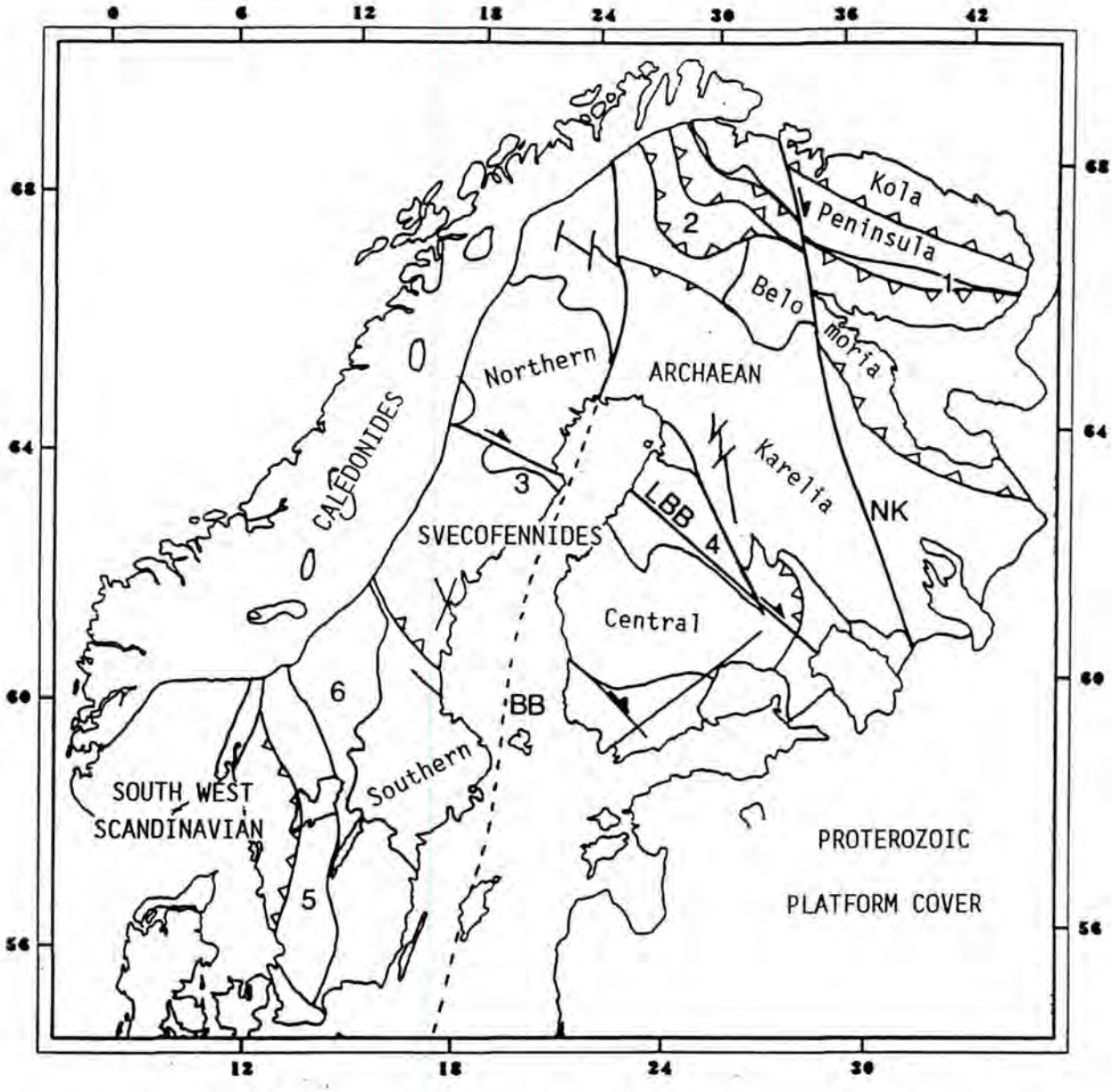
Advances in technology and investigative methods over the last two decades have brought geoscientists closer to understanding the complex geology of the Baltic Shield and its evolution. With increased understanding have come a succession of interpretations and evolutionary models.

The general evolutionary scheme adopted here is that put forward by Gaál and Gorbatschev (1987), which appears to have gained a wide acceptance in more recent publications (BABEL Working Group 1990, 1991; Sundblad 1991; Anderson 1991) although there are still some with alternative viewpoints (Welin, 1987, Witschard, 1984).

The model holds that some form of plate tectonics was active far back into the Archaean and can explain all the known features of the Baltic Shield crust. Five Precambrian orogenic cycles are identified :

Archaean	Saamian	(>3.1 - 2.9 Ga)
	Lopian	(2.9 - 2.6 Ga)
Proterozoic	Svecofennian	(2.0 - 1.75 Ga)
	Gothian	(1.75 - 1.5 Ga)
	Sveconorwegian	(1.25 - 0.9 Ga)

Most of the Baltic Shield crust was formed during the Lopian, Svecofennian and Gothian orogenies and can hence be divided into three major domains : Archaean (northeast) which also contains some older

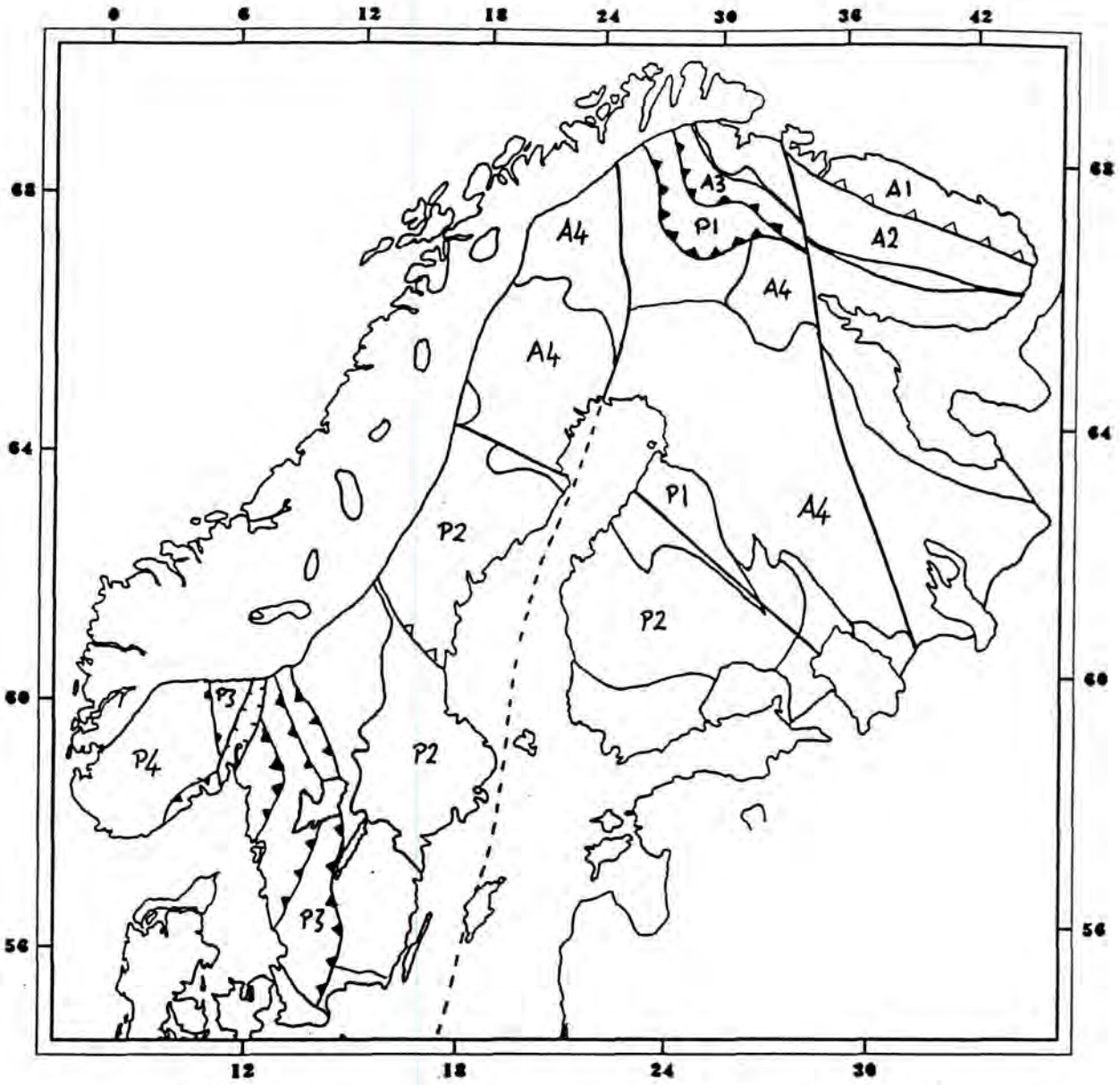


Key

1. Pechenga-Varzuga Belt
2. Lapland Granulite Belt
3. Skellefte Ore Zone
4. Vihanti-Pyhäsalmi Ore Zone
5. Protogine Zone
6. Transscandinavian Igneous Belt

NK North Karelian Shear Zone
LBB Ladoga-Bothnian Bay Shear Zone
BB Baltic-Bothnian Shear Zone

Fig. 2.1a)Tectonic map of the Baltic Shield, after Gaál and Gorbatschev (1987) and Berthelsen and Marker (1986).



Key

A1-A4	Units with	Archaean crustal ages
P1	Units with	Early Proterozoic crustal ages (~2.0-1.95 Ga)
P2	" "	Early " " " (~1.9-1.85 Ga)
P3	" "	Middle " " " (~1.7-1.65 Ga)
P4	" "	Middle " " " (~1.55-1.5 Ga)

Fig. 2.1 b) Tectonic units in the Baltic Shield based on both time of formation and time of most recent deformation, after Berthelsen and Marker (1986).

Saamian rocks, Svecofennian (central) and South West Scandinavian (Fig. 2.1a). Later orogenies merely deformed this crust with relatively little new crustal generation. Berthelsen and Marker (1986) point out that crustal units can be classified both according to time of original formation and also to time of most recent deformation. They combine these schemes to give a larger number of crustal units (Fig. 2.1b).

2.2 General overview

The major crustal developments are outlined here in chronological order.

2.2.1 Archaean

The Archaean domain consists of three provinces : the Karelian province is made up of granite greenstone belts of low metamorphic grade (Gaál, Mikkola and Söderholm, 1978) whilst in the Belomorian and Kola Peninsula domains high grade gneisses predominate (Gorbunov, Zagorodny and Robonen, 1985). The Proterozoic rocks of the Lapland granulite belt, which will be discussed later, separate the Kola Peninsula province from the rest of the Archaean domain.

The oldest known rocks in the Baltic Shield have been dated at 3.1 to 2.9 Ga old (Kröner, Puustinen and Hickman, 1981), being formed in the Saamian orogeny. They are found in Karelia, Lapland and the Kola Peninsula. Little more is known of this event and it is debatable whether coherent crust was formed over large areas or in smaller blocks which were later joined together (Gorbunov et al.,1985).

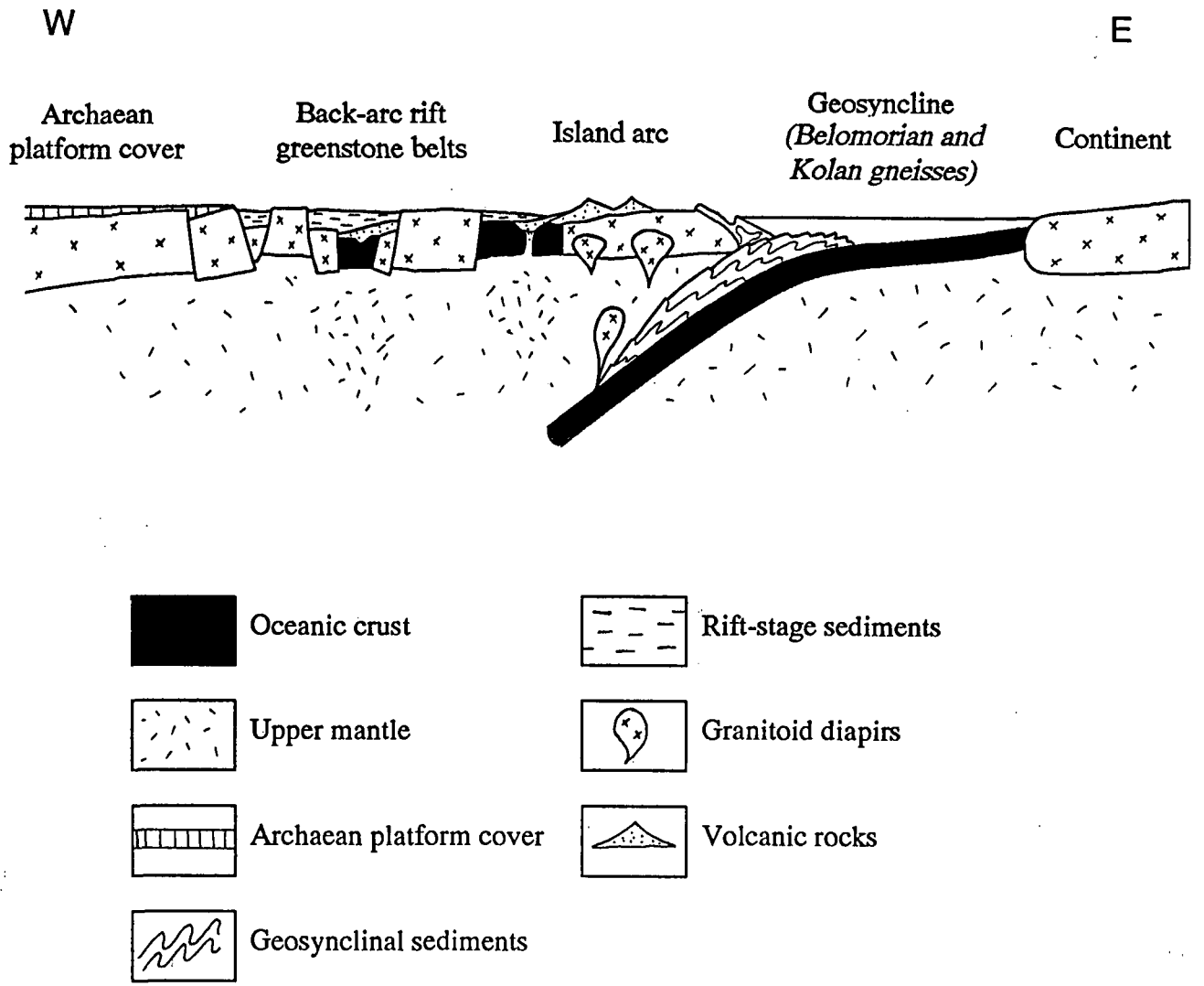


Fig. 2.2 A tectonic model of Lopian orogeny, after Gaál (1986)

The larger part of the Archaean crustal domain was formed during the Lopian orogeny (2.9 to 2.6 Ga ago). A plate tectonic mechanism for this is shown in Fig. 2.2 (Gaál, 1986). The granite-intruded Karelian greenstone belts are identified with the rifted foreland complex of a Saamian craton in the west, whilst the high grade gneisses of the Belomorian and Kola Peninsula provinces represent the mobile belt region and colliding continent in the east.

2.2.2 Late Archaean - Early Proterozoic cratonisation

The Lopian orogeny was followed by a long period of cratonisation lasting for over 500 Ma. During this period, the metasedimentary and volcanic rocks of the Lapponian Supergroup (2.7-2.4 Ga ago, Lapland) and the Sumian-Sariolan Group (2.5-2.3 Ga ago, further south in Finland and Russia) were laid down, controlled by NW trending faulting (Saverikko, 1983, Laajoki, 1986). The Karelian Province subsequently saw the thick platform cover of the Jatulian sedimentary sequence 2.3-2.1 Ga ago (Fig. 2.3). However this is absent in the Belomorian Province and is restricted to volcanic rock in fault-bounded basins in the Kola Peninsula (Gorbunov et al., 1985).

2.2.3 Svecofennian

At the end of this tectonically quiet period, intense rifting and dyke intrusion signalled the development of a passive continental margin trending NW at what is now the western edge of the Karelian Province (Gaál and Gorbatschev, 1987). Svecofennian crustal accretion proceeded

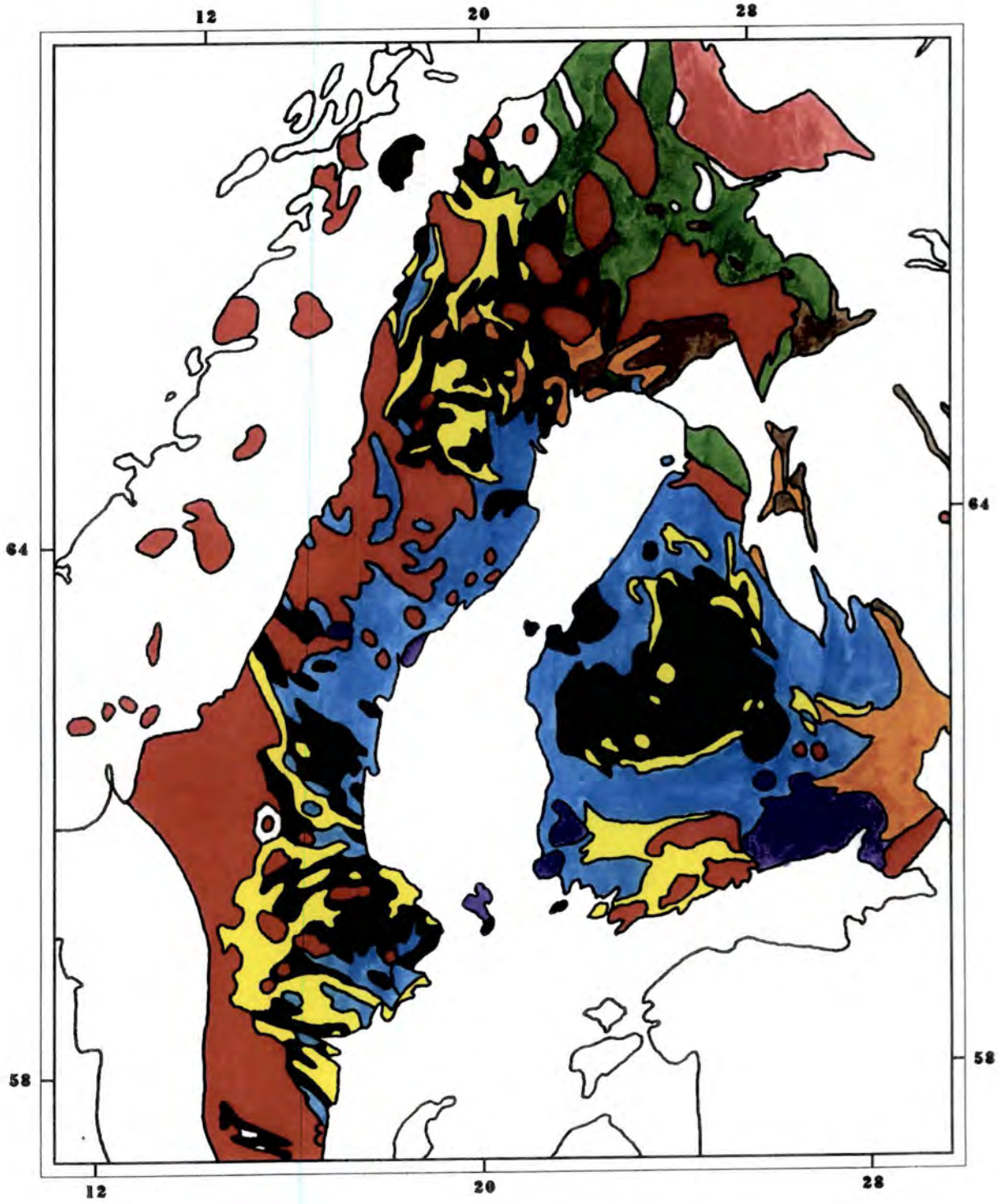


Fig. 2.3 Tectonostratigraphic map of the Svecofennides, after Gaál and Gorbatschev (1987) and Andersson (1991).

Episode 1 granitoids 1.90-1.86 Ga

Episode 2 granitoids 1.83-1.77 Ga

Episode 3 granitoids 1.65-1.56 Ga

Svecofennian metasediments

Svecofennian metavolcanics

Lapland Granulite Belt 2.1-2.0 Ga

Kalevian sediments 2.0-1.95 Ga

Jatulian and Sariolan sediments 2.5-2.1 Ga

Lapponian sediments 2.7-2.4 Ga

by sedimentation along this continental margin (Ward, 1987) and, after its activation, by volcanic activity and a series of plutonic episodes (Andersson, 1991). The bulk of the new crust was formed by intrusion of plutonic rocks into metasediments and metavolcanic belts on an unknown basement between 1.9 and 1.8 Ga ago. This occurred in an island arc environment after the subduction zone had moved to the southwest. Deformation and metamorphism throughout the Svecofennian Domain, and also in the Karelian Province, occurred synchronously with the major orogenic granitoid episodes (Lundberg, 1980, Welin and Stålhös, 1986).

The Lapland granulite belt and Pechenga-Varzuga belt consist of metamorphosed Proterozoic rocks and lie within the Archaean domain. They are considered to have been formed by closure of an oceanic basin 2.0-2.1 Ga ago, at the same time as the extensional phase in the Svecofennian Domain at the start of the Svecofennian orogeny. There is disagreement as to which belt formed the original ocean basin and to the direction of subduction (Barbey et al., 1984, Berthelsen and Marker, 1986).

The Pechenga-Varzuga belt is cut by the North Karelian shear zone, a deep strike-slip fault system with a dextral displacement of about 100 km initiated between 1.9 and 1.8 Ga ago (Fig. 2.1a). As the Svecofennian crust consolidated, another transcurrent fault system developed at the boundary between Archaean and Proterozoic crust: the Ladoga-Bothnian Bay fracture zone. This 50-100 km wide zone runs almost parallel to the previous plate margin and is dated at around 1.8 Ga ago. A further contemporaneous strike-slip shear zone, the Baltic-Bothnian Megashear, is proposed by Berthelsen and Marker (1986) to run roughly north from the Bothnian Bay and south beneath the Gulf of Bothnia and Baltic Sea. Although there has up to now been no geophysical evidence for this, it was

hoped that the BABEL experiment would provide the necessary information. These megashears are thought to have been active at least twice, in opposite senses, and to have caused offsets in the Moho.

2.2.4 Gothian and Sveconorwegian orogenies

Further granitic intrusion in the Svecofennian Domain, including the Rapakivi Granites, occurred in the period 1.65-1.56 Ga ago. These were thought to be anorogenic or related to the Gothian orogeny (Gaál and Gorbatschev, 1987) but more recent work associates them with the tail end of the Svecofennian orogeny (Andersson, 1991) which overlapped with the start of the Gothian. Most of the South West Scandinavian Domain crust was formed during the latter event (1.75-1.50 Ga ago). Sporadic intrusions continued until the Sveconorwegian orogeny (1.25-0.90 Ga ago) in which the SW Scandinavian crust was extensively reworked and intruded. The effects of the Sveconorwegian orogeny are bounded by the Protogine Zone, a region of deep faulting (Fig. 2.1a).

2.3 Svecofennides in more detail

The Svecofennian domain was formed in a series of alternating periods of sedimentation and igneous activity. The domain can be subdivided into three provinces with respect to the palaeoenvironment: a northern and a southern volcanic district are separated by a central marine sedimentary region known as the Bothnian Basin (Figs. 2.1a and 2.4, Welin, 1987). No basement rocks have been identified within the domain.

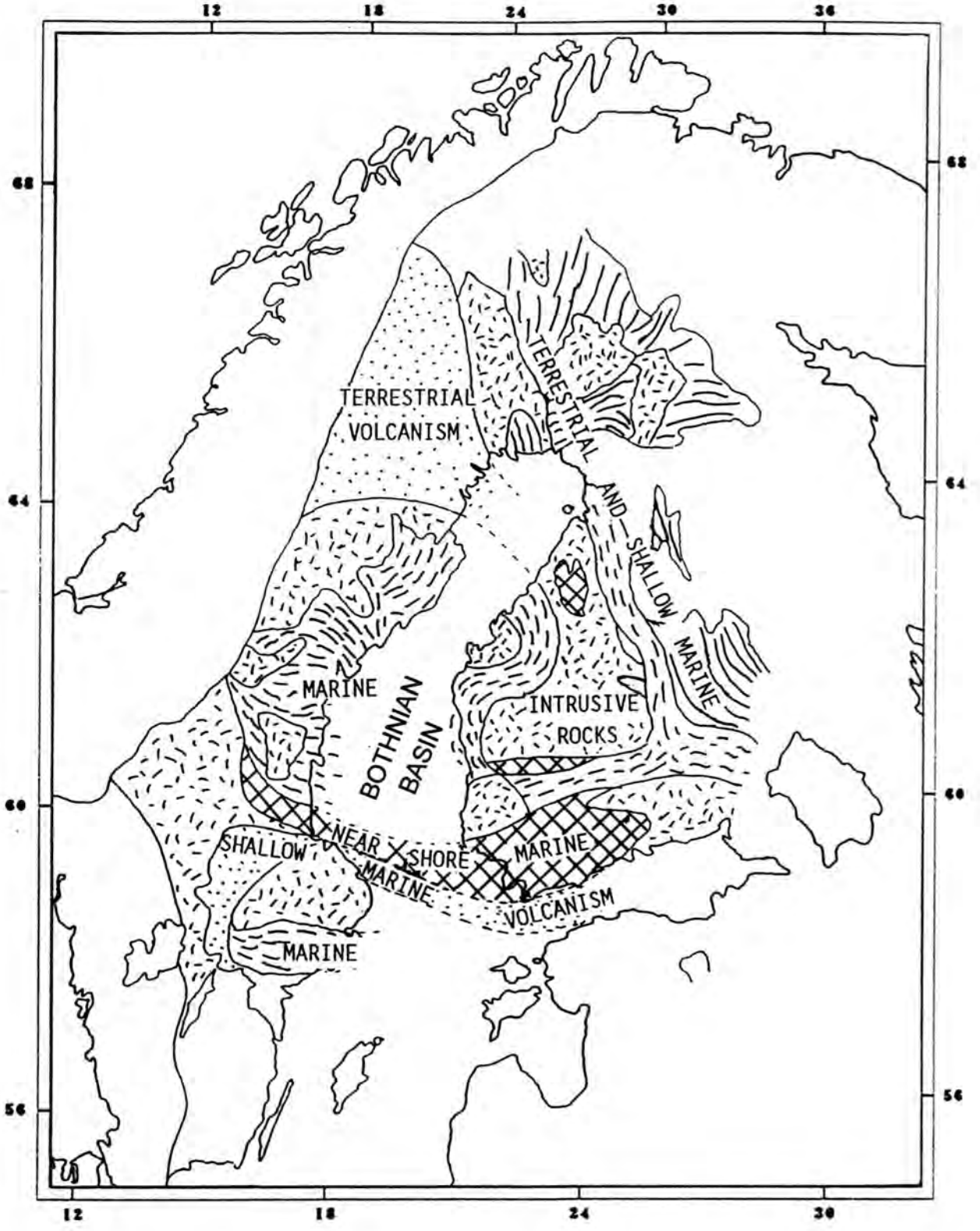


Fig. 2.4 Paleoenvironment map of the Svecofennian, after Welin (1987)

However, stratigraphic studies of the southern volcanic province indicate that Archaean crust may have been present to the southwest (Lundqvist, 1987).

Prior to 2.0 Ga ago, Jatulian sediments were being deposited in basins on the Archaean continent (Fig. 2.3). These were intruded by dykes and sills and unconformably covered by Kalevian turbiditic sediments derived from eroded Archaean and Jatulian bedrock when, between 2.0 and 1.95 Ga ago, a passive continental margin was formed. Displacement along the Ladoga-Bothnian Bay zone may also have been initiated at this time. The Kalevian sediments formed a 100 to 150 km wide belt along the new continental margin. As the marine basin deepened, sediments were carried further to the west to form the Svecofennian greywackes and pelites (Welin, 1987).

The Outokumpu district to the southwest of the Kalevian belt contains distal type turbidites in a separate marginal basin, and an association which has been contentiously interpreted as an ophiolite (Gaál and Gorbatschev 1987). This indicates the activation of the continental margin with subduction to the east, beneath the Archaean craton. The Skelleftea and Vihanti-Pyhasalmi ore zones are considered to be remnants of an early island arc system associated with this subduction which contributed little to the formation of new continental crust. Later, the subduction zone shifted a considerable distance to the southwest, initiating the main stage of Svecofennian crustal accretion in an island arc environment 1.9 Ga ago (Fig. 2.5).

Welin (1987) refutes the ophiolitic interpretation of the Outokumpu association and puts forward a block tectonic evolution as an alternative to the plate tectonic model while Witschard (1984) states the case for

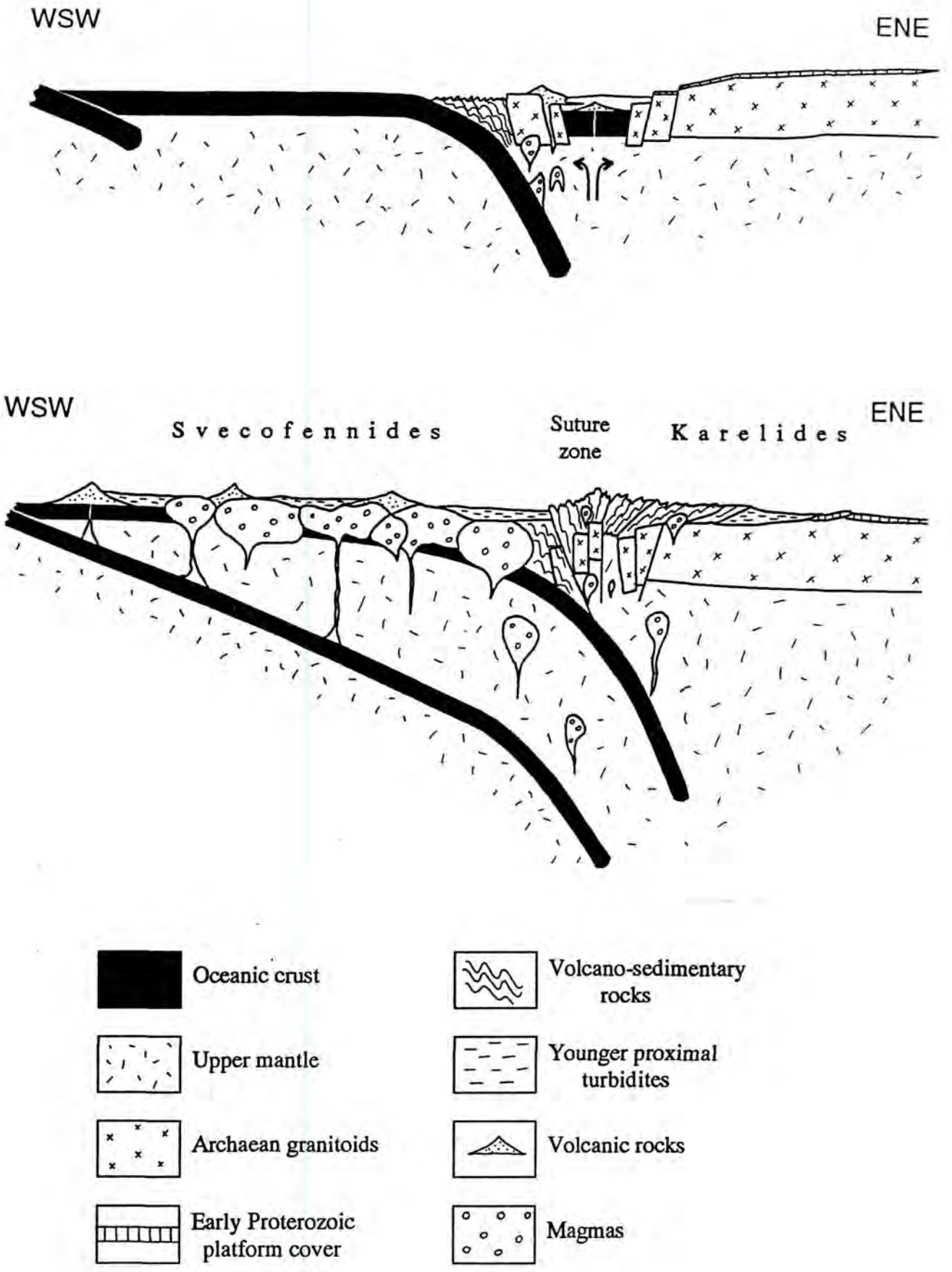


Fig. 2.5 Tectonic model of the Svecofennian orogeny :

a) 1920 Ga ago, b) 1880 Ga ago. (After Gaál, 1986)

basement reactivation. However, although the Outokumpu association has now been shown to be merely the product of primitive igneous activity in a pelagic basin (Park, 1988), a true ophiolite has been discovered at nearby Jormua (Kontinen, 1987). The plate tectonic view is backed up by seismic evidence of dipping reflectors in the lower crust beneath the Bothnian Bay (BABEL Working Group, 1991).

Three major plutonic episodes occurred during the Svecofennian orogeny (Andersson, 1991). The first involved the formation of large volumes of new crust between 1.90 and 1.86 Ga ago. The shallow submarine and subaerial volcanics of the Northern and Southern Svecofennian Provinces (1.90-1.88 Ga) are associated with this episode. They were little affected by large scale fault movements and formed the northern and southern boundaries to the marine basin of the Central Province. Most of the crust formed was juvenile, mantle-derived material although the component of remelted Archaean crust increases to the north, near the Archaean craton. The sedimentary and volcanic processes terminated in the marine basin at around 1.86 Ga ago (Welin, 1987). There followed a quiescent interval interrupted only by a small amount of rifting and dyke intrusion with minimum metamorphism until the second Svecofennian plutonic episode began 1.83 Ga ago. This time only minor quantities of juvenile crustal material were introduced, the main body being made up of remelted episode-1 material. Numerous massifs and dyke networks were intruded in the two volcanic provinces and larger masses emplaced in the former central sedimentary basin and the Archaean Domain of Finnish Lapland. Also included in this group is the Transscandinavian Igneous Belt, a huge batholith forming the southwestern boundary of the Svecofennian domain. The second episode

ended at around 1.77 Ga ago as the main tectonic activity moved west of the Svecofennian craton.

Cratonisation of the Svecofennides with little igneous activity preceded the third episode of granitoid generation (1.65-1.56 Ga ago). The Rapakivi Granites were formed during this period of intracratonic reworking, along with mafic dyke swarms associated with extensional tectonism. A thick sequence of red arkose Jotnian sandstone was deposited over a large area around 1.3 Ga ago but is only preserved locally in deep down-faults and grabens and in the Gulf of Bothnia. The sandstones are intruded by 1.25 Ga old dolerite sills and dykes.

2.4 Paleozoic sedimentary geology of the Bothnian Sea

Although few deposits of Paleozoic sedimentary rocks are to be found around the coastline of the Bothnian Sea, a considerable sedimentary sequence exists in the central and western parts of the Sea (Winterhalter, 1972, Fig. 2.6).

The Svecofennian igneous and metamorphic crystalline basement which predominates around the coast is continuous from the eastern side beneath the Sea, with a shallow sea floor gradient (Fig. 2.9) and no evidence of boundary faulting along the coast (Fig. 2.6). The western side is bounded by strong down-faulting (a). Large areas are covered by Jotnian sandstones which give the first indication of a post-Svecofennian marine environment. Their thickness is unknown over most of their extent due to the similarity in seismic velocity to the basement rocks although boreholes give thicknesses between 500 and 1000 m. The Jotnian rocks have been

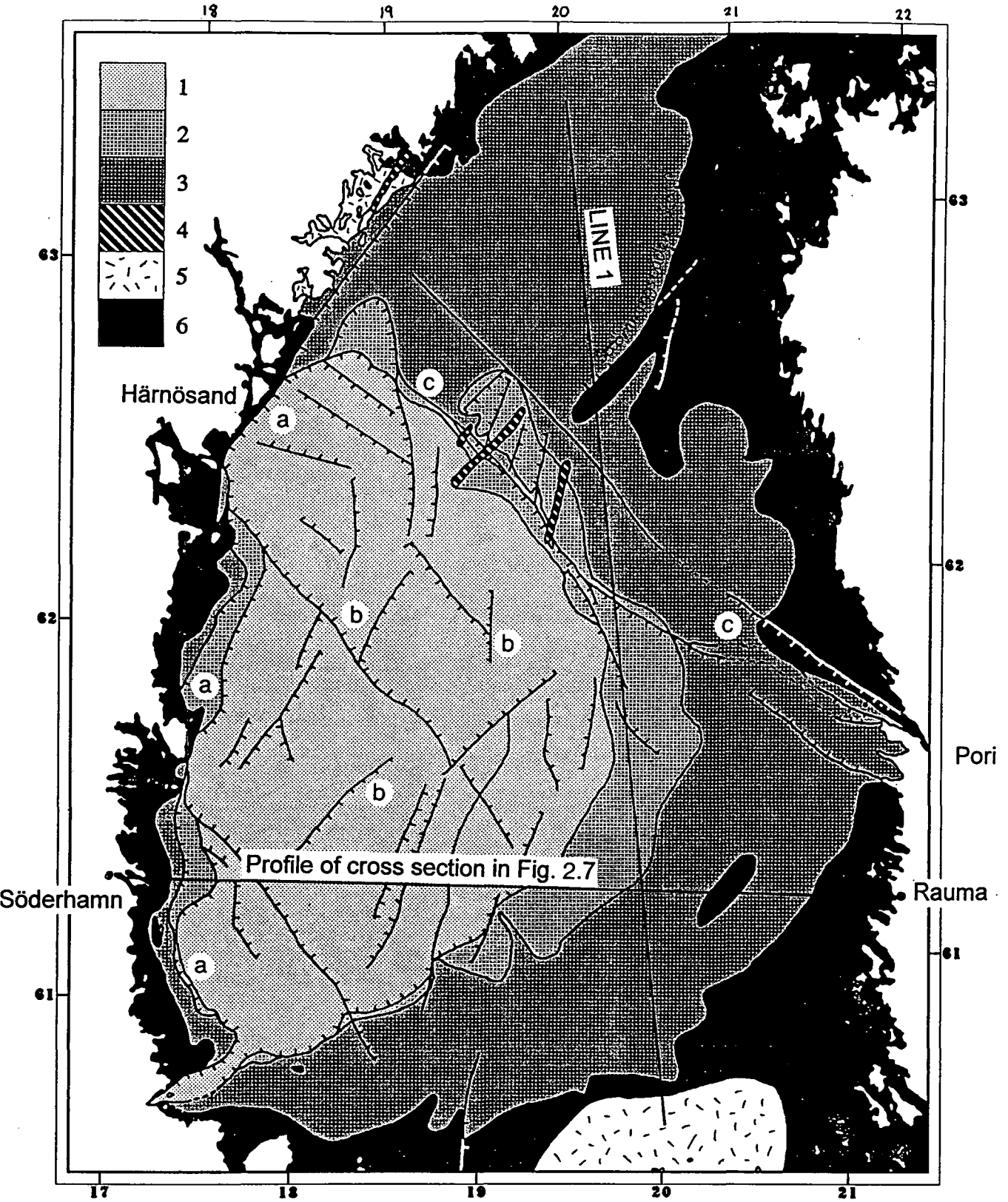


Fig. 2.6 Geology map of the Bothnian Sea, after Ahlberg (1986)

1 Ordovician; 2 Cambrian; 3 Jotnian sst.; 4 Diabase intrusion;
5 Younger granites; 6 Precambrian crystalline basement

a-c : faults - see text.

considerably faulted and intruded by dolerite dykes and sills during subsequent tectonic activity. After a period of erosion, the sub-Cambrian peneplain underwent a transgression of the sea giving the Cambrian sandstones and clays before the Lower Ordovician shales and Middle and Upper Ordovician limestones were deposited. The shale-limestone boundary is a strong seismic reflector. Fig. 2.7 shows an east-west cross-section of the sea bed across the southern part of the Sea from Söderhamn to Rauma, while Fig. 2.8 shows an isopach map of Paleozoic sediment thicknesses. The sediments have remained for the most part undisturbed but evidence for post-depositional faulting and folding has been observed in central parts of the formations (Fig. 2.6,b) and a large fracture zone runs roughly NW from Pori to Härnösand (c). The current sea basin is of Tertiary age and was produced by intense down-faulting on the western side. Faulting also occurred within the Sea itself.

Tectonic activity was renewed with the glacial loading of the Quaternary. The net effect of glaciation was an accumulation of sediments rather than erosion. The mean thickness of post-glacial sediments is around 1.9 m.

Seismic velocities of the major sea floor formations are as follows (per. com. G. Lindsey, Bullard Laboratories, Cambridge, 1991)

Quaternary till	2120 m s^{-1}
" clay	2500 m s^{-1}
Ordovician limestone	3400 - 4100 m s^{-1}
Jotnian sandstone	4780 m s^{-1}
Crystalline basement	5640 - 6230 m s^{-1}
	(mean 5850 m s^{-1})

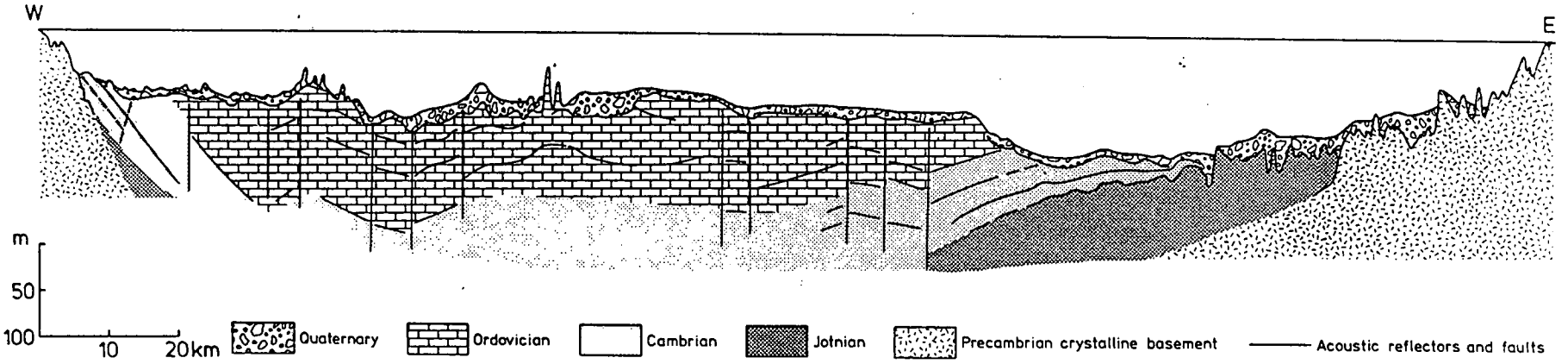


Fig. 2.7 Schematic geological profile (W—E) across the southern part of the Bothnian Sea from Söderhamn in Sweden to Rauma in Finland. (After Flodén and Winterhalter, 1981).

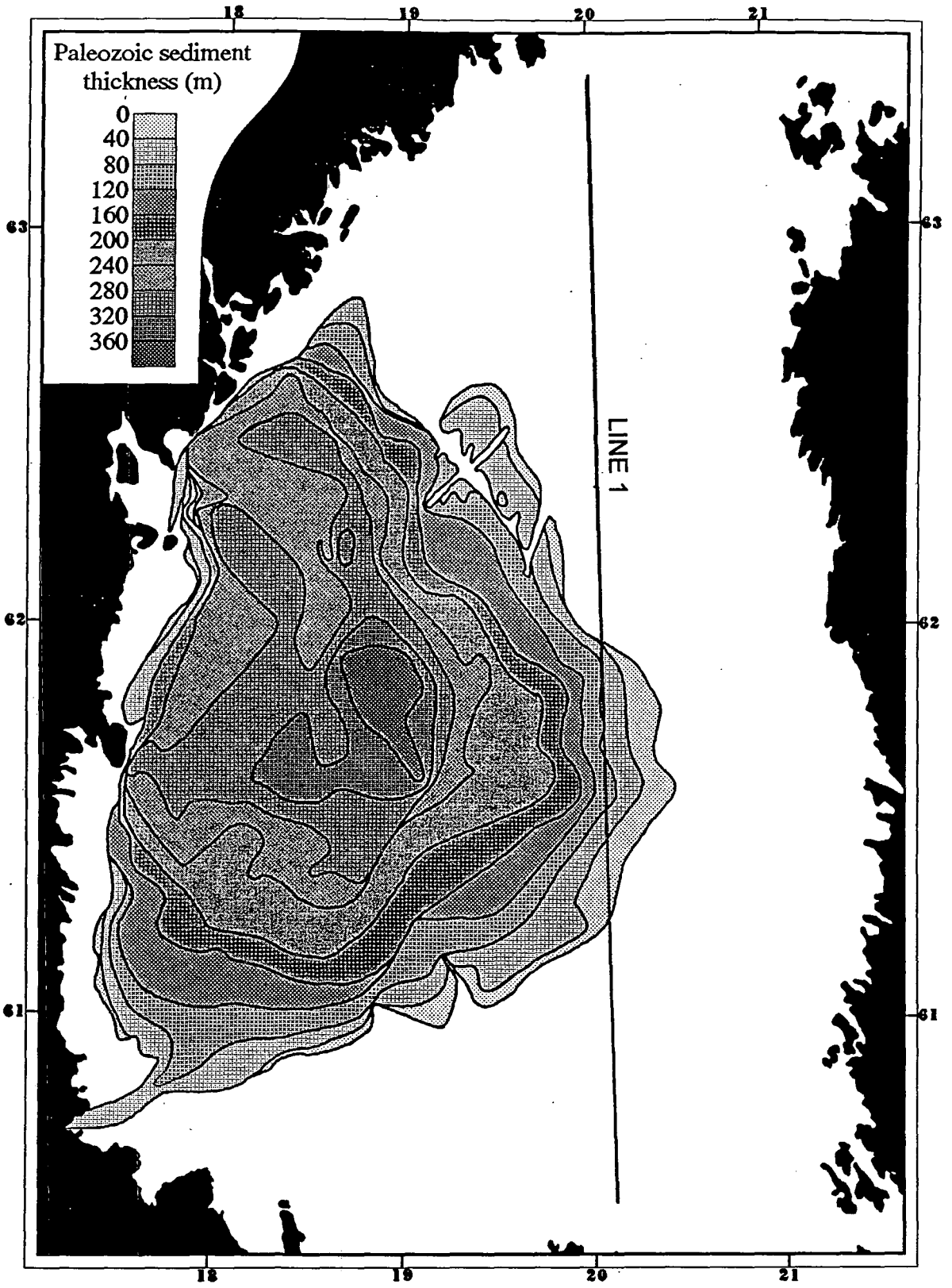


Fig. 2.8 Paleozoic sediments isopach map of the Bothnian Sea.

after Axberg, 1981.

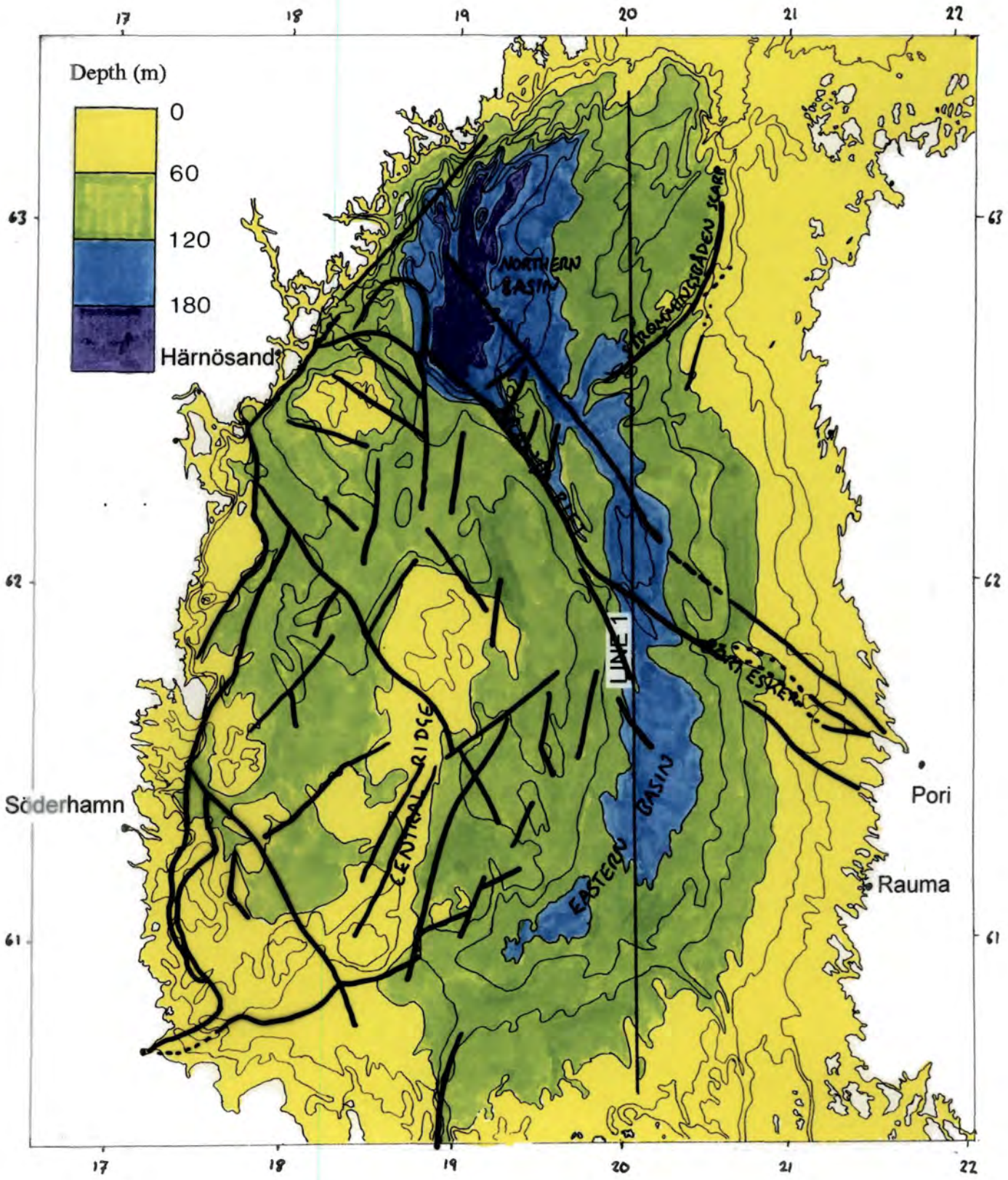


Fig. 2.9 Bathymetry map of the Bothnian Sea, after Winterhalter (1972).

The sea floor itself exhibits a wide variation in depth, although the mean water depth is only 68 m (Fig. 2.9). The low relief of the Finnish coast extends westwards with uniform shallow gradient into a deep region of the Sea known as the Eastern Basin. The western slope of the basin is much more irregular, rising up to the Central Ridge which is a broad shallow region running N-S in the southwestern part of the Sea. The Eastern Basin is bounded to the North by the junction of the Pori Esker (a narrow glacial outwash deposit) and a narrow, faulted valley called the Aranda Rift. This widens to the northwest and deepens into the Northern Basin, the deepest part of the Sea. In the Northeast, the Strömmingsbåden scarp system runs for about 100 km and reaches 170 m in height at its southern end where it trends SW. Fig. 2.10 shows an echo sounding profile across the scarp in an E-W direction while Fig. 2.11 shows the variation of water depth along BABEL Line 1, as recorded by the survey ship. The overlay for Fig. 2.9 shows that most of the bathymetric features mentioned here are bounded by faults. Winterhalter (1972) claims that most of these faults are of Tertiary age, associated with the formation of the Sea, but some may be older.

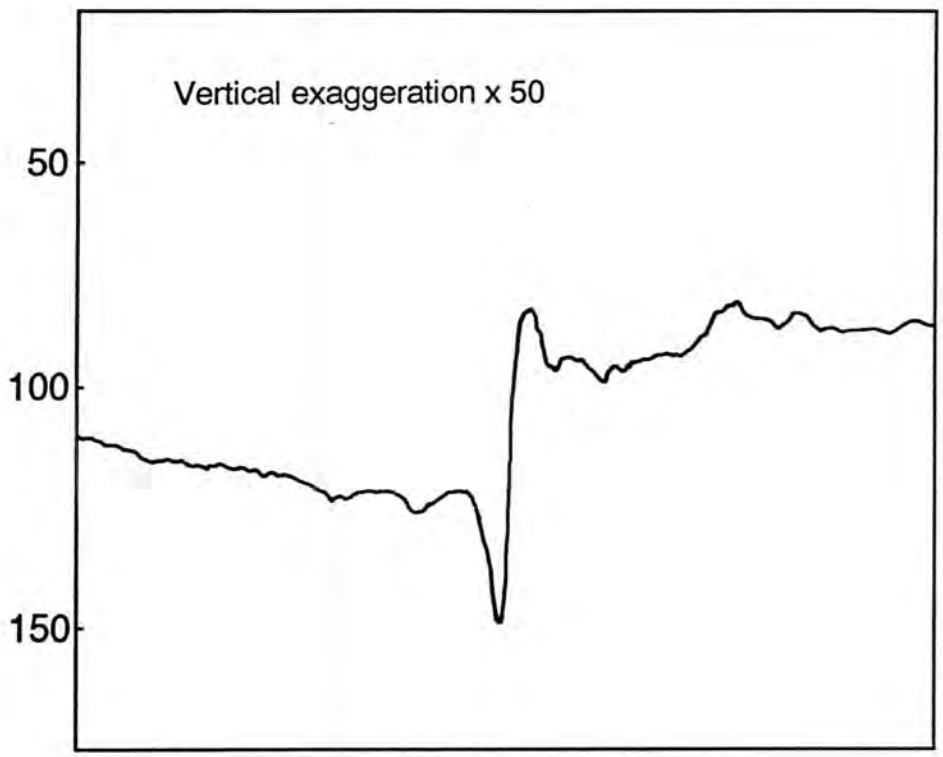


Fig. 2.10 E-W echo sounding profile of the Strömmingsbåden scarp.
(After Winterhalter, 1972).

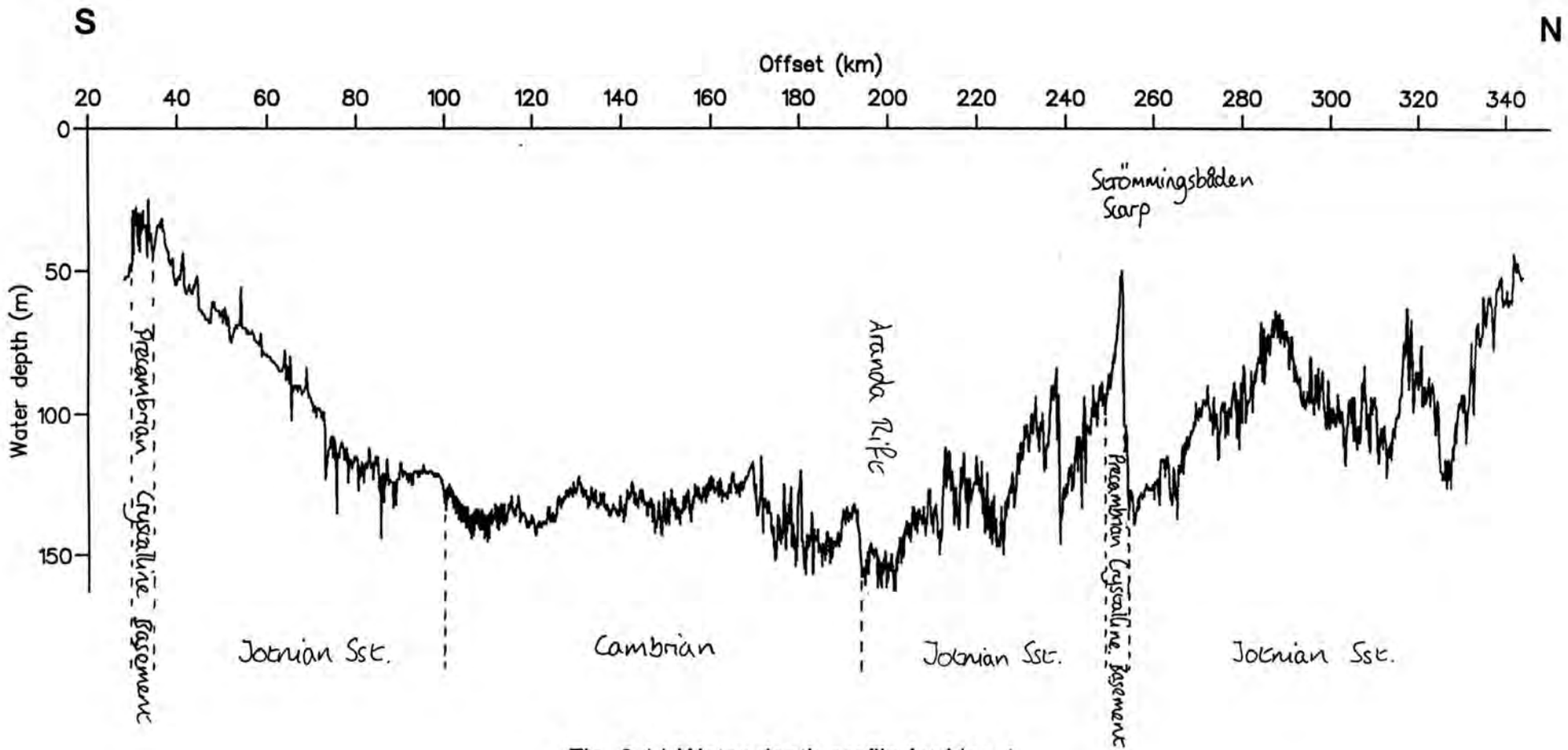


Fig. 2.11 Water depth profile for Line 1.

3. Previous Geophysical Surveys

3.1 Seismic surveys

A number of wide angle deep seismic surveys have been carried out within the Baltic Shield over the last thirty years. Those in and around the Bothnian Sea region of the Svecofennides are shown in Fig. 3.1. A summary of the acquisition details of these are given in Table 3.1. All the surveys used large explosive charges (generally underwater in lakes or the sea) with recording stations on land. With the exception of the Sylen-Porvoo line, a small number of sources were shot into a larger number of receivers at a few kilometres spacing and the lines were reversed. The spatial resolution is relatively low compared to the BABEL survey. The few previous surveys over Precambrian shield crust which match BABEL's resolution include the GLIMPCE project in the Great Lakes of North America (shot spacing 50-300 m; GLIMPCE seismic refraction working group, 1989) and the marine parts of the EUGENO-S project of Northern Germany and Southern Scandinavia (shot spacing 300 m; EUGENO-S working group, 1988). Fig. 3.2 shows sample data sections from the EUGENO-S project between Denmark and Sweden and from shot E of the FENNOLORA profile to the west of, and roughly parallel to BABEL Line 1.

The 2D models produced for the five profiles from the Sea of Bothnia are given in Fig. 3.3a-e. From these it can be seen that both low and high velocity zones have been proposed in the upper crust. In the SVEKA survey (Fig. 3.3b), a thick, high velocity lower crustal layer has been modelled.

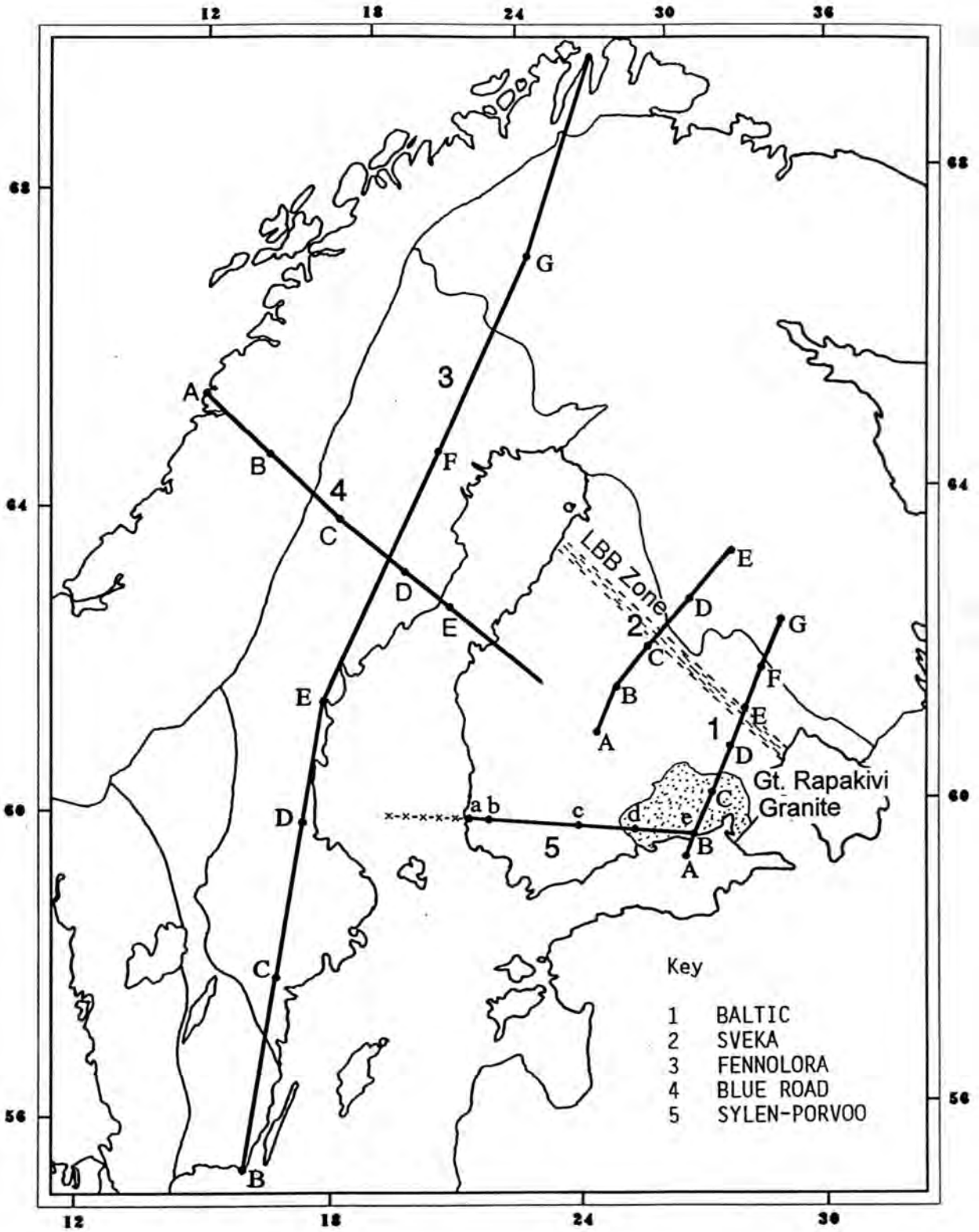
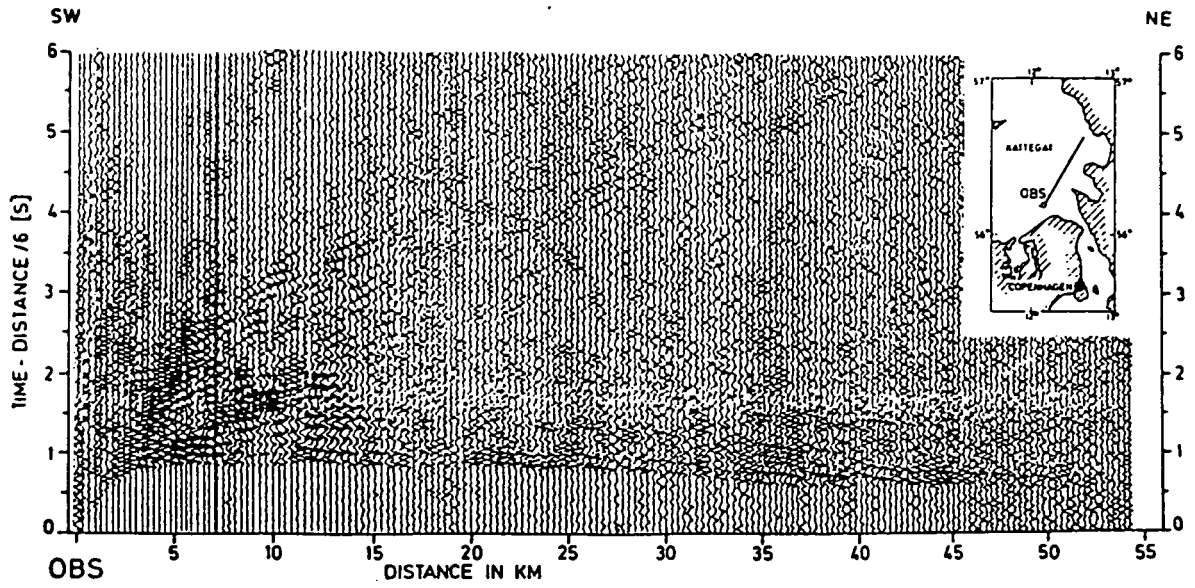


Fig. 3.1. Map of previous seismic surveys, after Luosto, 1990

Profile	Date	Shot separation	Receiver separation	Reversed	Source type
BALTIC	1982	80 km	2 km	Y	explosive 200-1700 kg
SVEKA	1981	80 km	2 km	Y	explosive 100-1000 kg
FENNOLORA Shots D and E	1979	180 km	3 km	Y	explosive 1800-8200 kg
BLUE ROAD	1972	110 km	4 km	Y	explosive 700-1400 kg
SYLEN-PORVOO	1965	3.5 km	60 km	N	explosive 50-300 kg
BABEL Line 1	1989	75 m	350 km	Y	marine airgun 120.6 l

Table 3.1 Seismic Profiles in the Baltic Shield



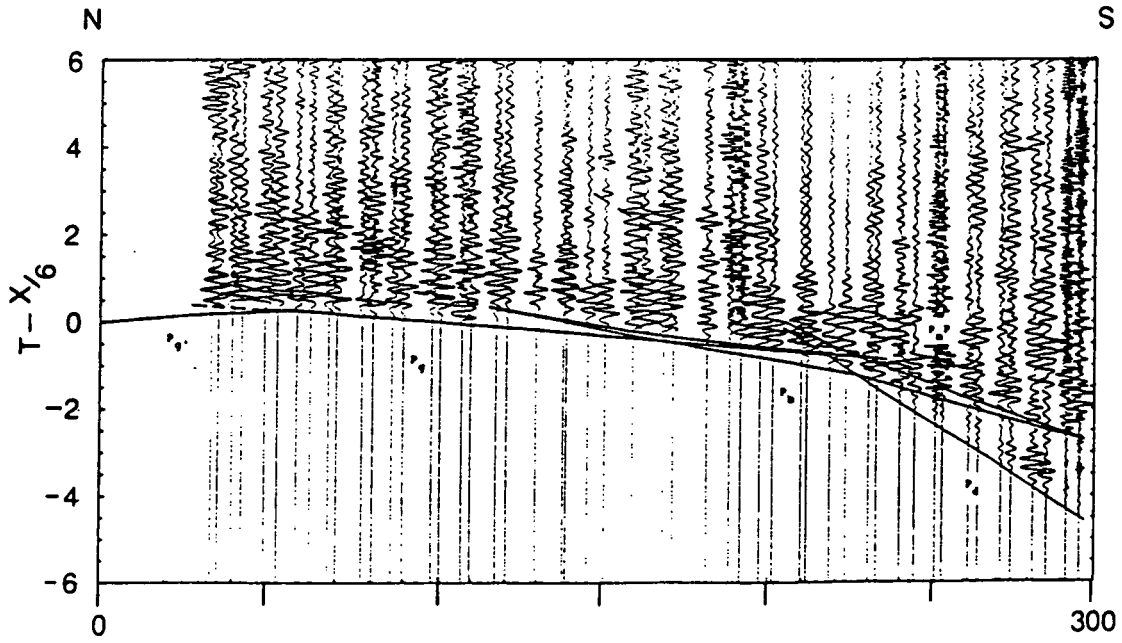


Fig. 3.2 Examples of data from earlier projects
a) EUGENO-S marine airgun data, OBS 1 (EUGENO-S working group, 1988)
b) FENNOLORA shotpoint E south: typical of experiments in the region (Guggisberg et al., 1991)

Perhaps the most dramatic features appear to be the deep depressions in the Moho boundary observed in the FENNOLORA, BALTIC and SVEKA profiles. In the latter two these are directly beneath the Ladoga-Bothnian Bay zone (Figs. 2.1 and 3.1).

3.1.1 BALTIC 1982 (Fig. 3.3a)

This survey, shot in southeast Finland, crosses the Ladoga-Bothnian Bay zone. The Moho is at about 40 km depth at the southern end, below the great Rapakivi granite intrusion. It deepens northwards and disappears in the centre of the profile where the crust extends down to a second boundary, elsewhere within the upper mantle, at a depth of 65 km. The P-wave velocity beneath this boundary is 8.5 km s^{-1}. In the northeast the crust thins again to about 45 km.

There is a high velocity body in the south between 10 and 14 km (V_P = 6.5 km s^{-1}) with lower velocities below down to about 18 km. A mid-crustal reflector known as the *c-boundary* has been widely observed in the Baltic Shield. It is defined as the crustal boundary below which the P-wave velocity exceeds 6.5 - 6.6 km s^{-1} (Luosto, 1990) and in this case varies between 18 and 22 km.

3.1.2 SVEKA 1981 (Fig. 3.3b)

Located in central Finland, this profile also crosses the Ladoga-Bothnian Bay zone. In this region the crust is modelled as being about 60 km thick, thinning to either side. The upper crust contains a low velocity zone across the whole profile and a locally present high velocity body while

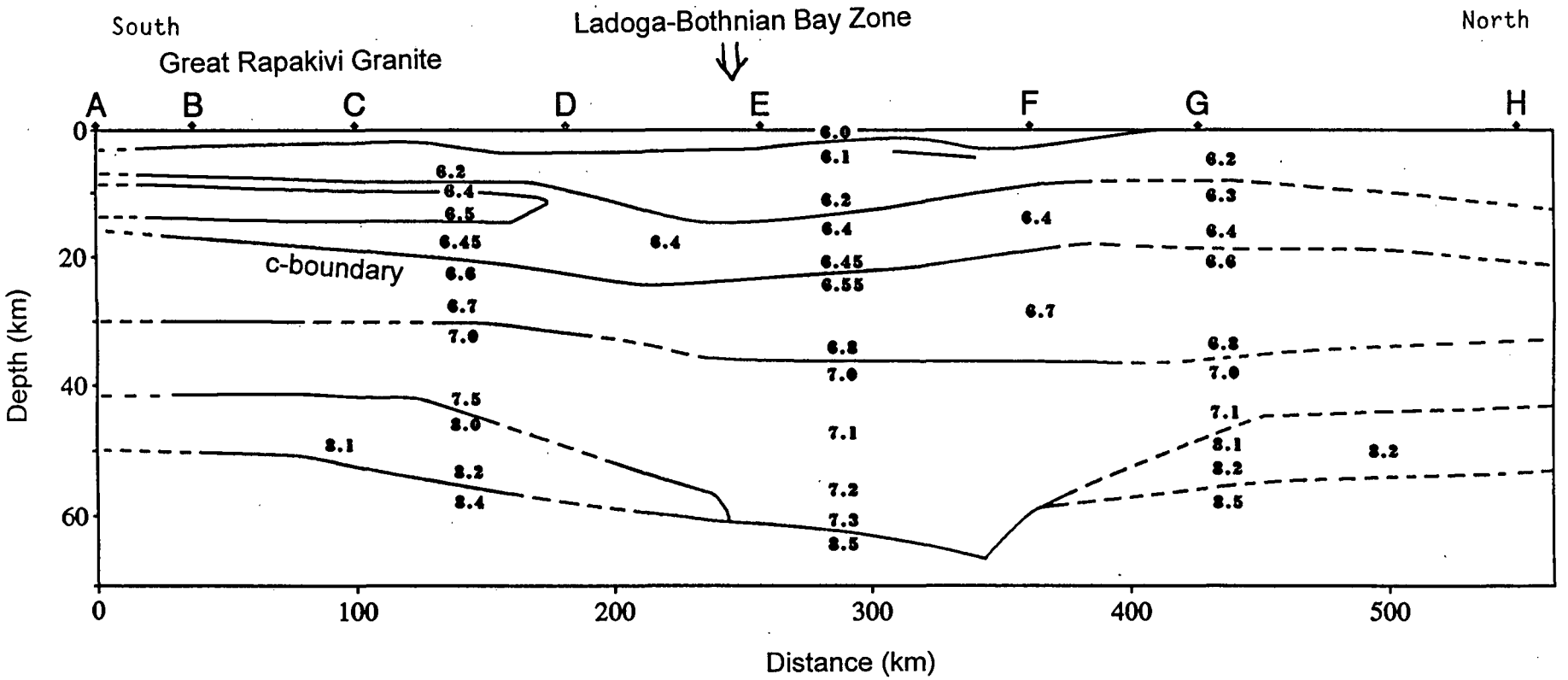


Fig. 3.3. Crustal velocity models from previous surveys:

a) BALTIC 1982 (after Luosto et al., 1990)

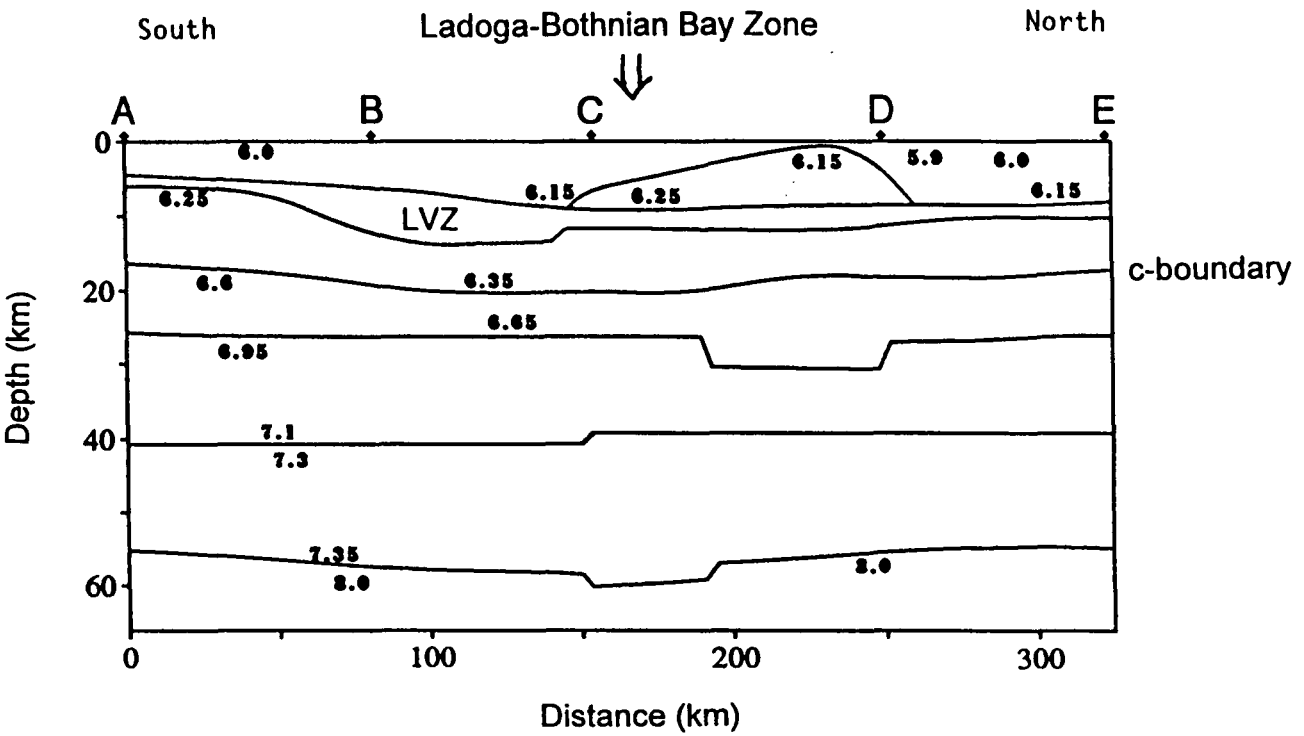


Fig. 3.3. b) SVEKA 1981 (after Grad and Luosto, 1987)

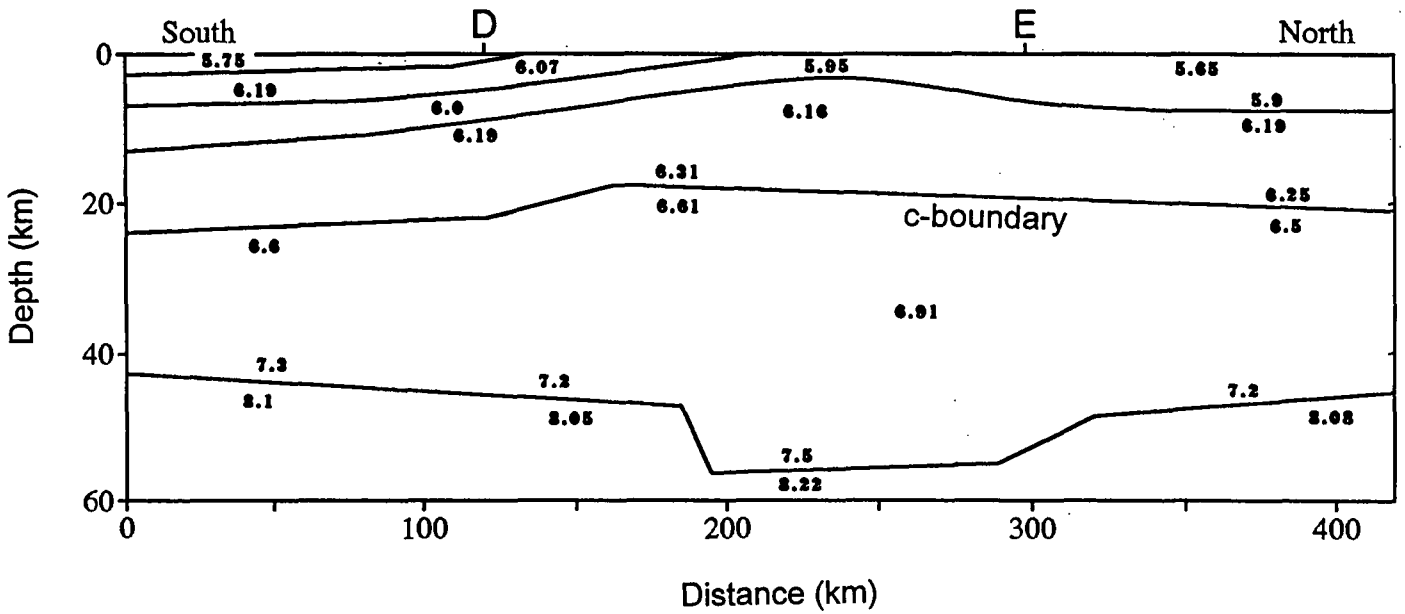


Fig. 3.3. c) FENNOLORA 1979 (after Guggisberg et al., 1991)

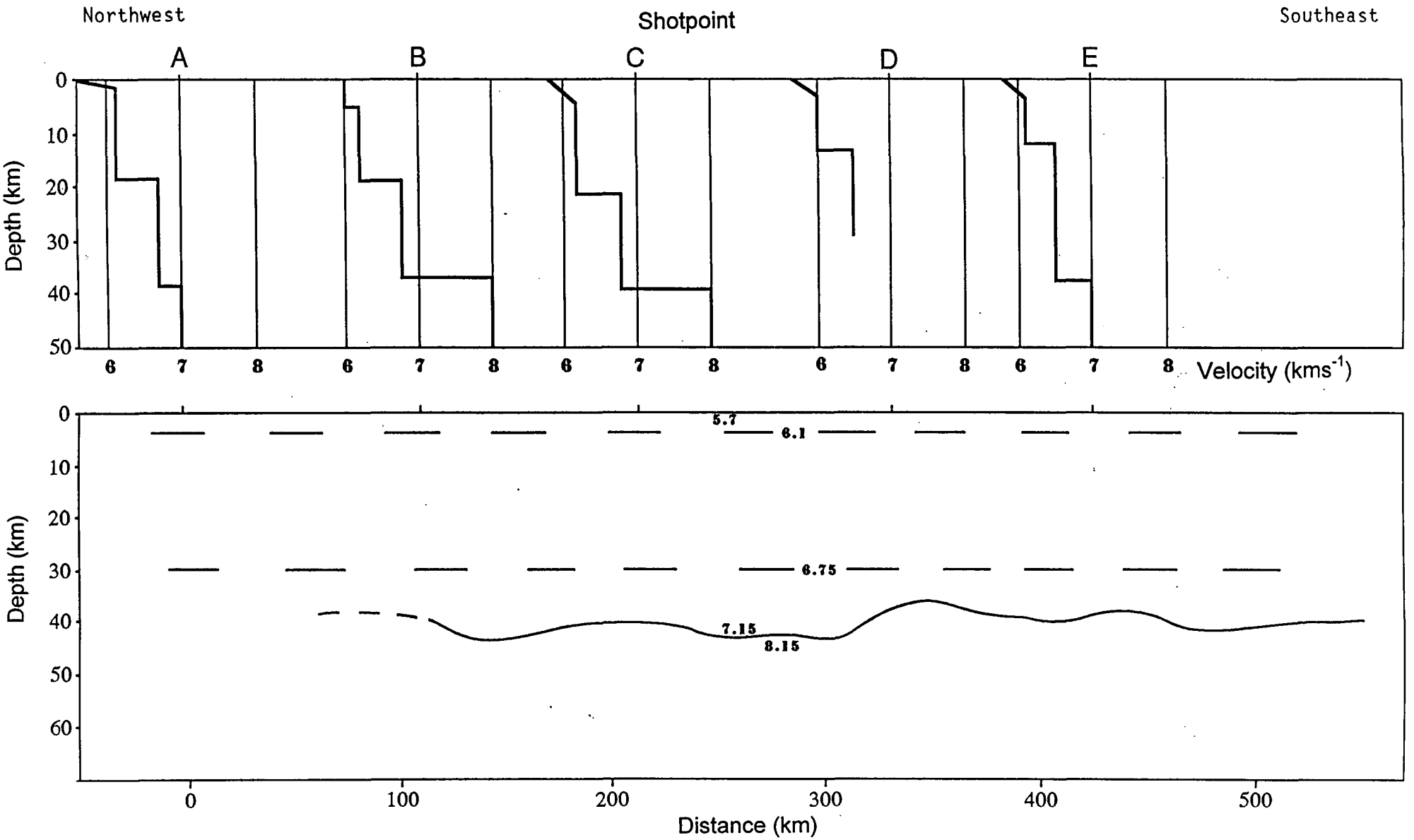


Fig. 3.3. d) BLUE ROAD 1972 (after Cassel and Fuchs, 1979; Lund, 1979)

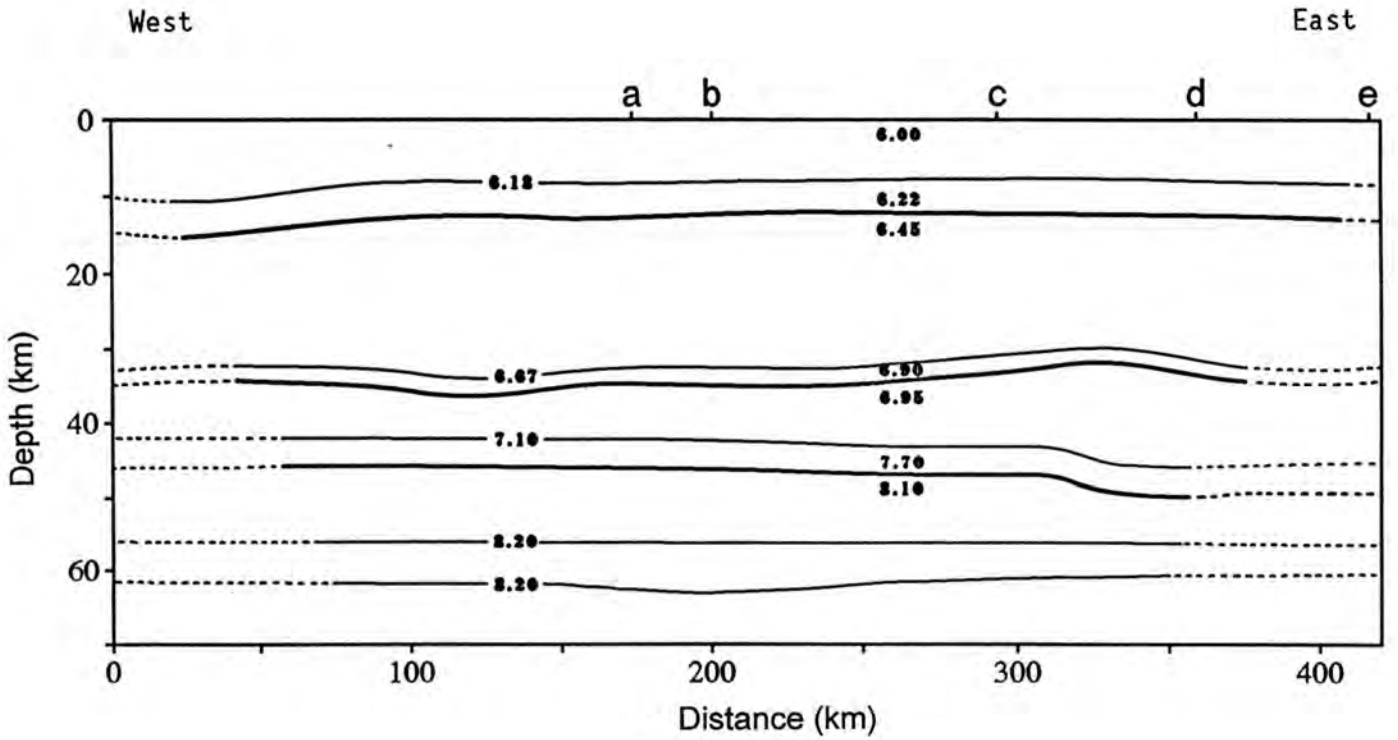


Fig. 3.3. e) SYLEN_PORVOO 1965 (after Luosto, 1986)

Heavy lines indicate velocity discontinuities while light lines are velocity isolines.

in the lower crust there is a 20 km thick layer with V_P greater than 7.3 km s^{-1}. A distinct c-boundary is observed at depth 17 - 20 km with a velocity discontinuity of 6.35 to 6.6 km s^{-1}.

3.1.3 FENNOLORA 1979 (Fig. 3.3c)

The FENNOLORA profile was shot as part of the European Geotraverse (EGT) project and covers a distance of over 2000 km from southern Sweden to northern Norway. The model shown in Fig. 3.3c is from the central section of the line between shotpoints D and E, adjacent to the Sea of Bothnia.

A depression of the Moho to a depth of about 55 km is seen in this region. The P-wave velocity within this depression is over 7.5 km s^{-1}. A similar feature occurs between shotpoints B and C to the south, and a shallower one around shotpoint G to the north.

The c-boundary was found throughout the profile with a velocity contrast of around 6.3 - 6.6 km s^{-1} at a depth varying between 16 and 24 km.

3.1.4 BLUE ROAD 1972 (Fig. 3.3d)

The Blue Road survey extends from the Norwegian coast near the Arctic circle, across Sweden and into central Finland. The figure shows 1D models from each shotpoint but secondary reflected arrivals were rarely seen on the record sections so a gradational model was preferred. The Moho was modelled using wave-front methods but boundaries within the crust could not be defined. The data has also been used to argue that

there is a series of alternating high and low velocity layers in the upper mantle down to a depth of about 100 km (Lund, 1979; Cassel and Fuchs, 1979).

3.1.5 SYLEN-PORVOO 1965 (Fig. 3.3e)

The data from this survey were originally acquired in 1965 but reinterpreted by Luosto (1986) using modern ray tracing methods. A line of marine explosive shots was fired in the Sea of Bothnia into a number of earthquake stations in southern Finland. The resulting low speed paper recordings did not provide very accurate data.

The Moho depth is 44 km for most of the model, increasing to 49 km in the east. Above this lies a 4 km thick transition zone with velocity increasing from 7.1 to 7.7 km s^{-1} and, as in the Blue Road profile, there appears to be a low velocity layer in the upper mantle between 56 and 63 km depth.

Reflections are seen from depths of 9 km and 44 km in the western part of the model which do not correspond well with the velocity layer boundaries.

3.1.6 Summary

Moho and c-boundary depths from these and other Baltic Shield surveys outside the region of interest have been compiled by Luosto (1990) to produce two contour maps (Figs. 3.4 (a) and (b)). The major features and trends can be clearly seen but detail cannot be relied upon due to the sparsity of data.

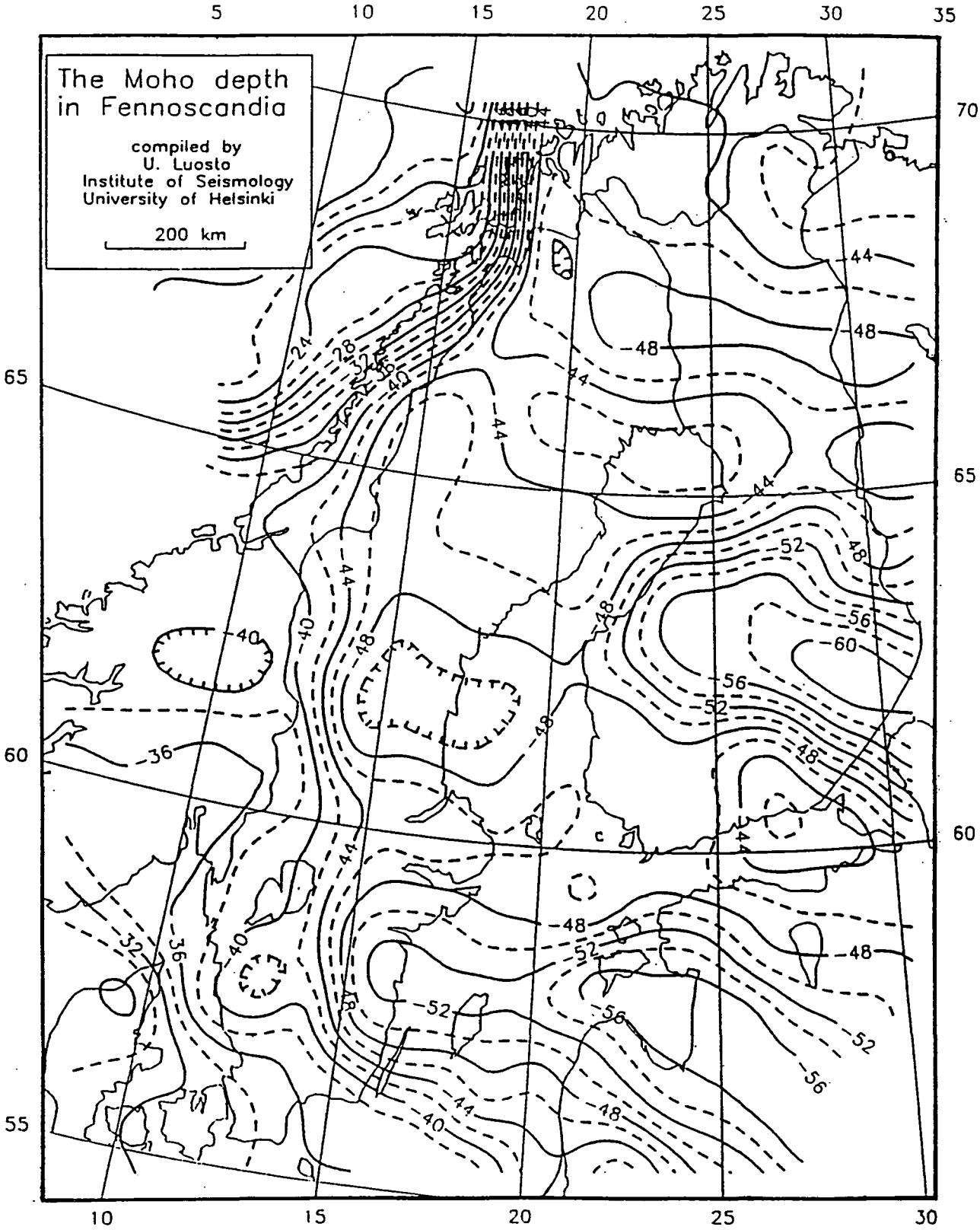


Fig. 3.4. a) Moho depth contour map (Luosto, 1990)

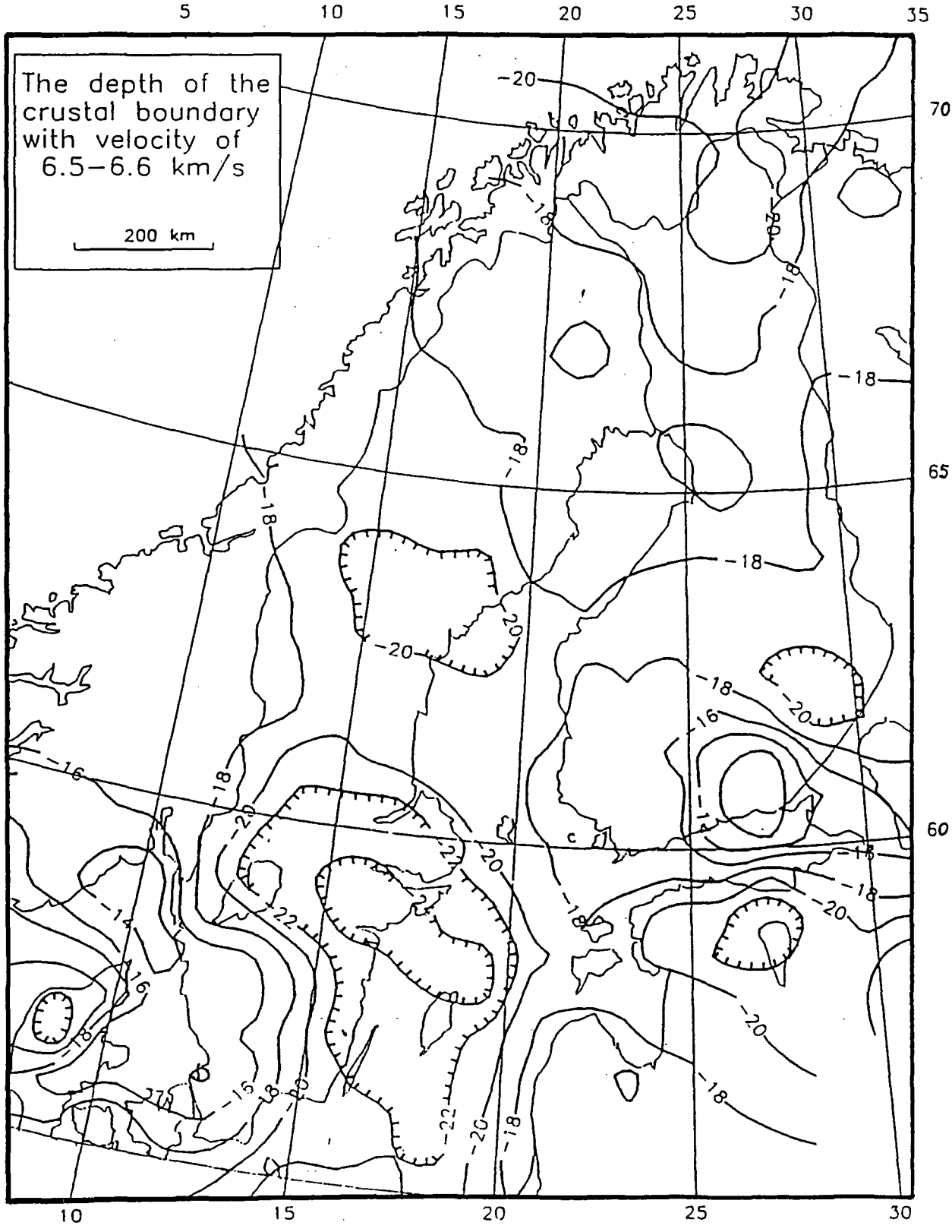


Fig. 3.4. b) c-boundary contour map (Luosto, 1990)

In the Moho map (Fig. 3.4a), the deepest depression is centred in eastern central Finland and extends to the north west, running roughly parallel to the Ladoga-Bothnian Bay zone. It is truncated at the Finnish coast of the Bothnian Bay. A second depression lies further to the south, in the Baltic States. This curves westwards across the Baltic sea into southern Sweden. A smaller depression occurs around the western coast of the Bothnian Sea, running WNW-ESE. The crust thins gradually to the north, south and west, and more steeply along the north west coast of Norway.

The c-boundary (Fig. 3.4b) has depressions in east central Finland and northern Estonia with a large high in between. There is a deep depression in south east Sweden and a shallower one in central Sweden, just north of the Moho depression. The c-boundary over the rest of the region appears to be relatively flat. The pattern of highs and lows in the c-boundary reflects that of the Moho topography described above. However, the features are not quite coincident, i.e. the highs and lows of the c-boundary do not lie directly above those of the Moho.

3.2 Gravity and aeromagnetic data

Gravity and magnetic data are available for the Sea of Bothnia and surrounding area. Fig. 3.5 shows a Bouguer anomaly contour map from Seasat and Geos3 altimetry data, grid spacing 30 km. There is a major gravity low in the northern Sea of Bothnia which seems to be part of a NE-SW trending band. A gravity high of similar trend but lesser lateral extent lies directly to the south. There appears to be no relationship between

these features and the c-boundary contour map but the band of negative gravity anomaly lies approximately on the line between the northwest tip of the Moho depression in central Finland and that in central Sweden. However, since there was no deep seismic data previously available from this area, the Moho contours here are not well defined. The only geological feature in the vicinity which has a similar trend is the southwestern end of the Strömmingsbåden Scarp system and its associated faults (Figs 2.6 and 2.9).

Fig. 3.6 is a contour map of high altitude (3000m) total magnetic field intensity, grid spacing 35 km. A large positive anomaly lies to the northwest of the Åland Archipelago and a lesser one in southern Finland. These may be associated with Episode 3 (Rapakivi) granite intrusions (Fig. 2.3) but are not quite coincident. They appear to be unrelated to any of the other geological or geophysical features already discussed.

A certain amount of 2½ D gravity and magnetic modelling has been carried out along BABEL lines 1 and 7 in parallel with this project (Pedersen, Tryggvason, Schmidt and Gohl, 1992). These models will be reviewed in the discussion in Chapter 8.

3.3 Conductivity data

Several magnetotelluric surveys have been carried out in Finland, the most relevant ones to this study being along the SVEKA wide angle seismic profile (see section 3.1) and near Oulu (Fig. 3.7, Hjelt, 1990). Though there is little direct correspondence between seismic and conductivity models and any given conductivity value cannot be associated

with a specific seismic velocity, one can compare the two types of model by looking for similarities in the size, shape and location of features. Because the SVEKA conductivity profile extends much further to the southwest than the seismic profile, 0 km on the distance scale in Fig. 3.3b corresponds roughly to 150 km in Fig. 3.7b. The depression in the boundary at 25-30 km depth towards the northeast of the seismic model corresponds approximately to the conductive zone at similar depth to the northeast of the Ladoga-Bothnian Bay (LBB) zone in the conductivity model. The region of higher upper crustal velocity directly above this in the seismic model (between shotpoints C and D in Fig 3.7b) may be related to the same feature. The deep conductive layer in the conductivity model (30 - 60 km depth) does correspond directly to the lower crust in the seismic model, there being much greater variation in depth in the conductivity model, especially at the northeastern end. However, the general trend of thicker crust in the central part of the seismic model (beneath the LBB zone) is echoed in the conductivity profile. On the basis of this, there is a case for suggesting that the crust is also thickened to the southwest, beneath the Tampere Schist Belt conductivity anomaly.

Fig. 3.8 shows the surface projections of crustal conductivity anomalies (Hjelt, 1992) while Fig. 3.9 shows an interpretation of magnetovariational surveys by Pajunpää (1987), dividing the crust into blocks based on changes in conductivity.

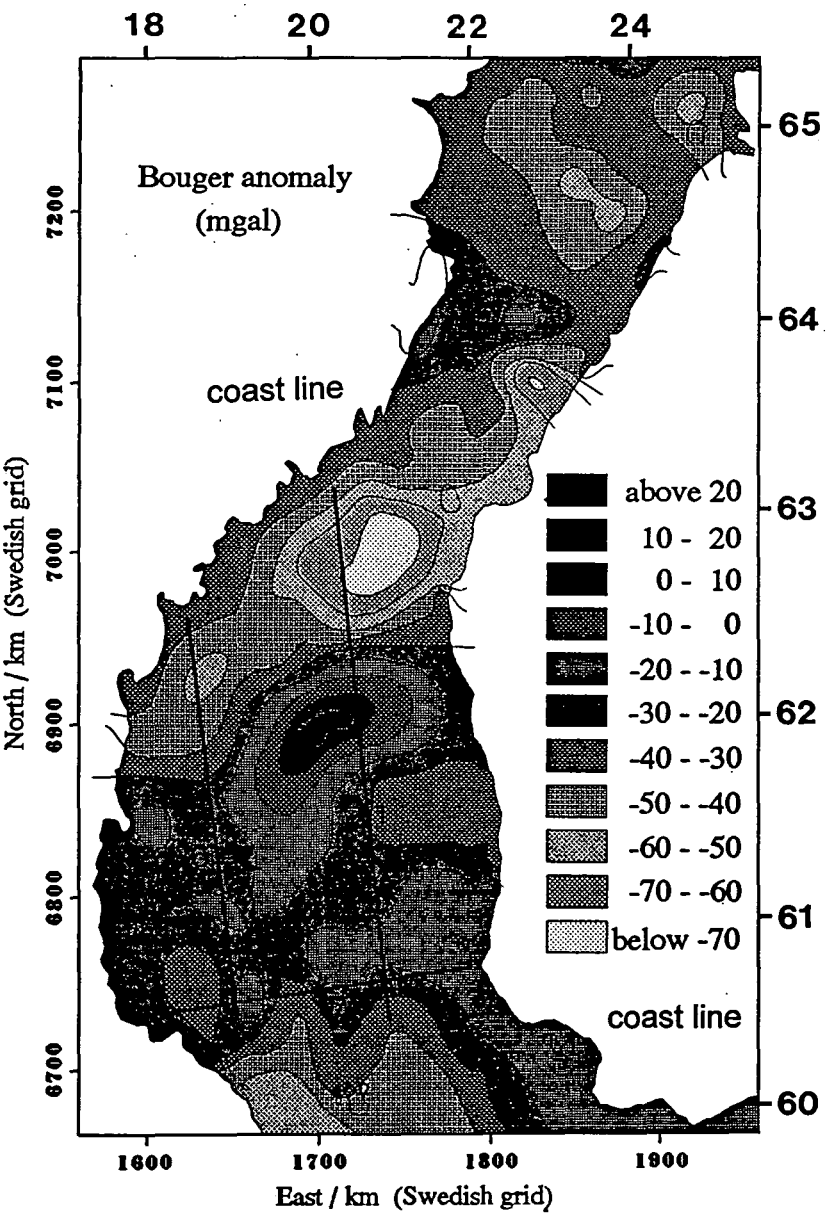
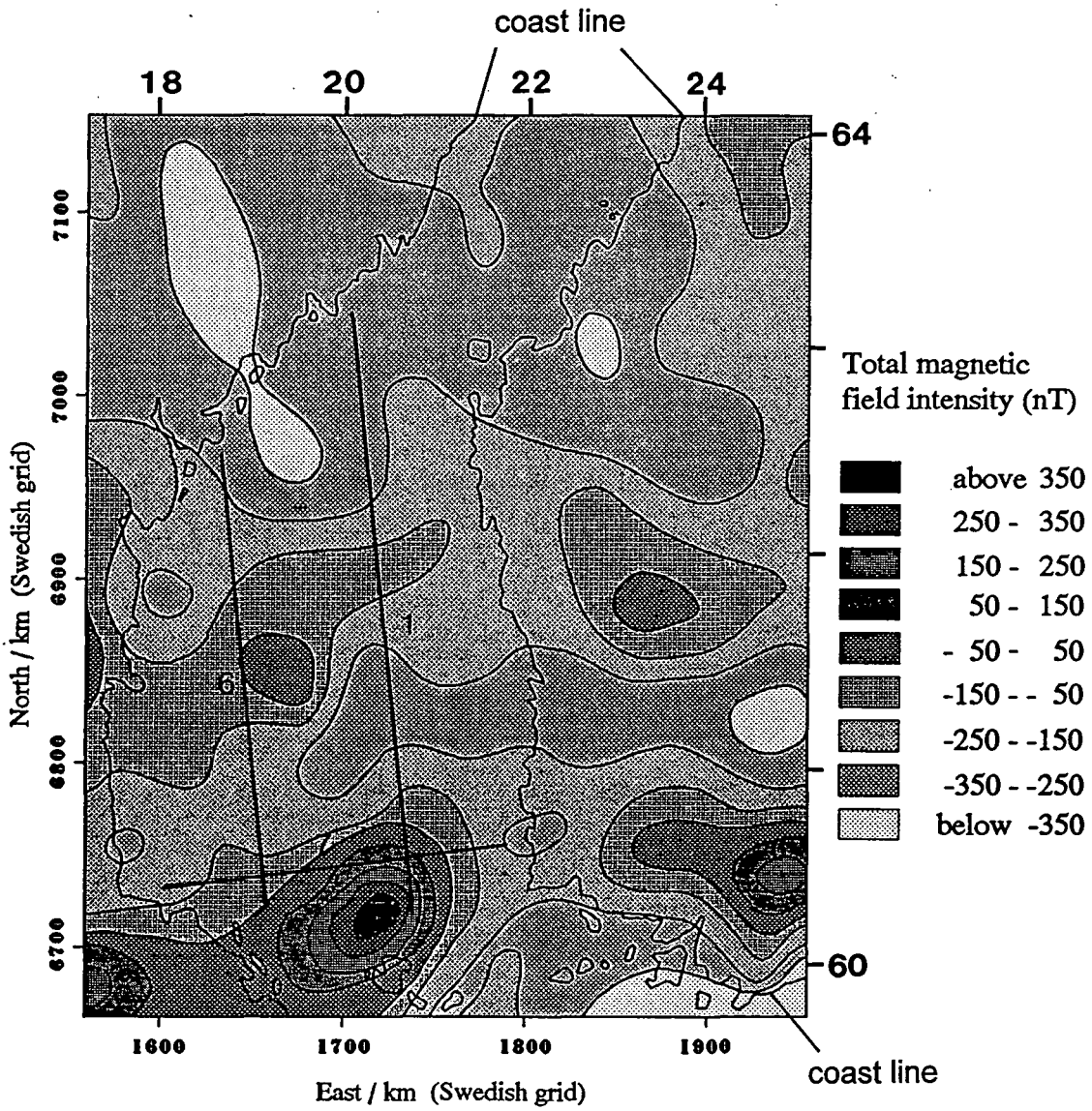
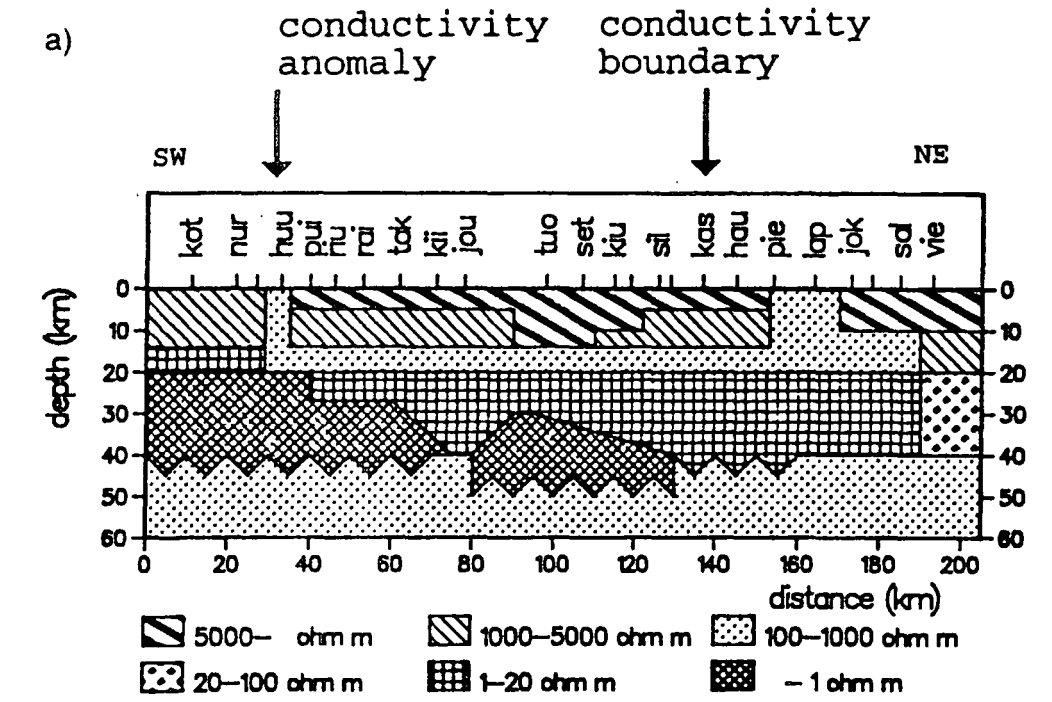


Fig. 3.5 Bouguer anomaly map of the Sea of Bothnia (after Pedersen et al., 1992).

Fig. 3.6 Aeromagnetic total intensity map of the Sea of Bothnia, altitude = 3000 m (after Pedersen et al., 1992)





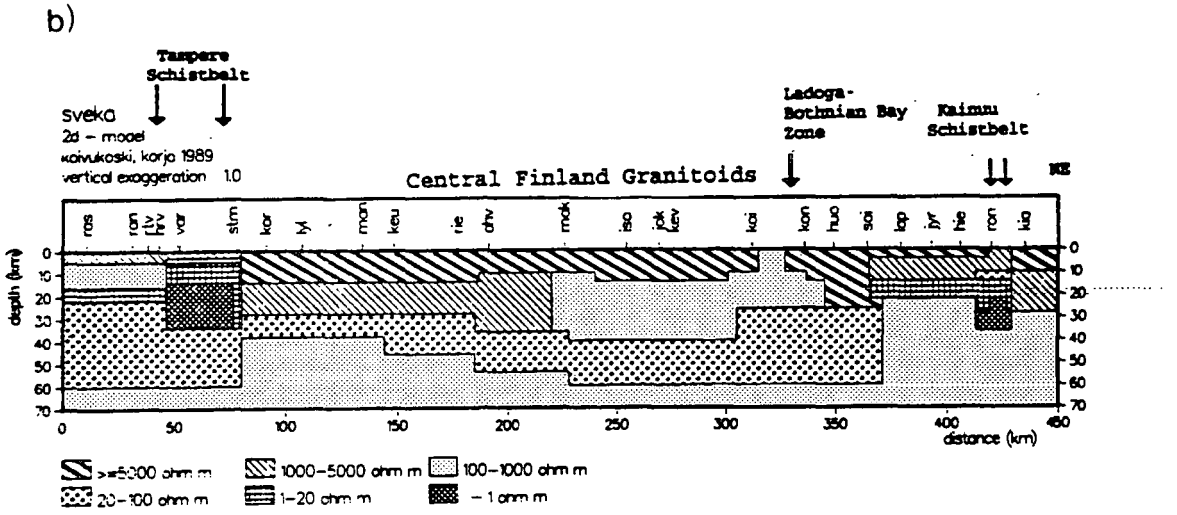


Fig. 3.7 Magnetotelluric conductivity models (Hjelt,1990):
a) Oulu IV profile
b) SVEKA profile

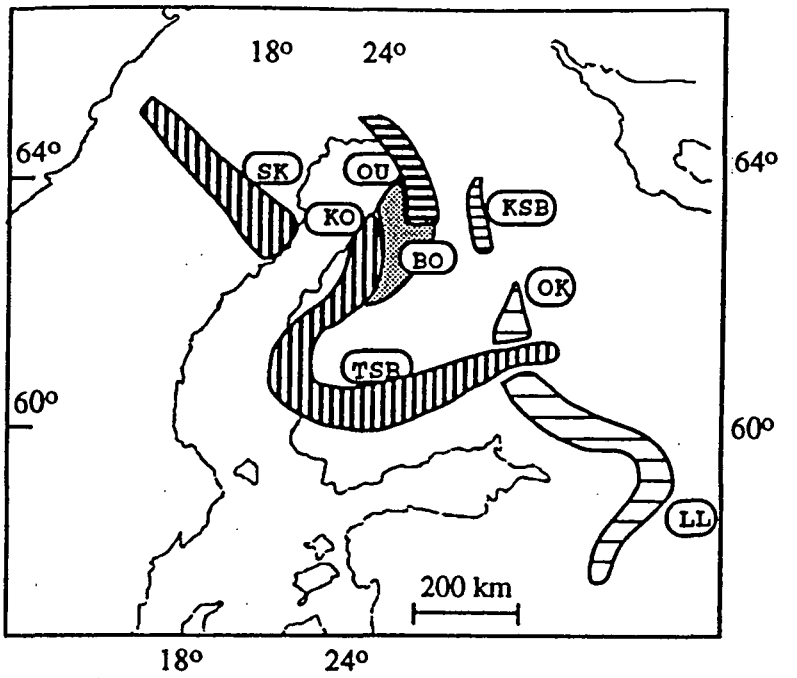


Fig. 3.8 Surface projections of crustal conductivity anomalies from magnetovariational studies (Hjelt,1992)

Conductors of the BB area: BO = Bothnian; KO = Kokkola; OU = Oulu; SK = Skellefteå-Storavan anomalies. Others: KSB = Kainuu Schist Belt; LL = Lake Ladoga; OK = Outokumpu; TSB = Tampere Schist Belt anomalies. The latter is a part of the SFCA = South Finland Conductivity Anomaly.

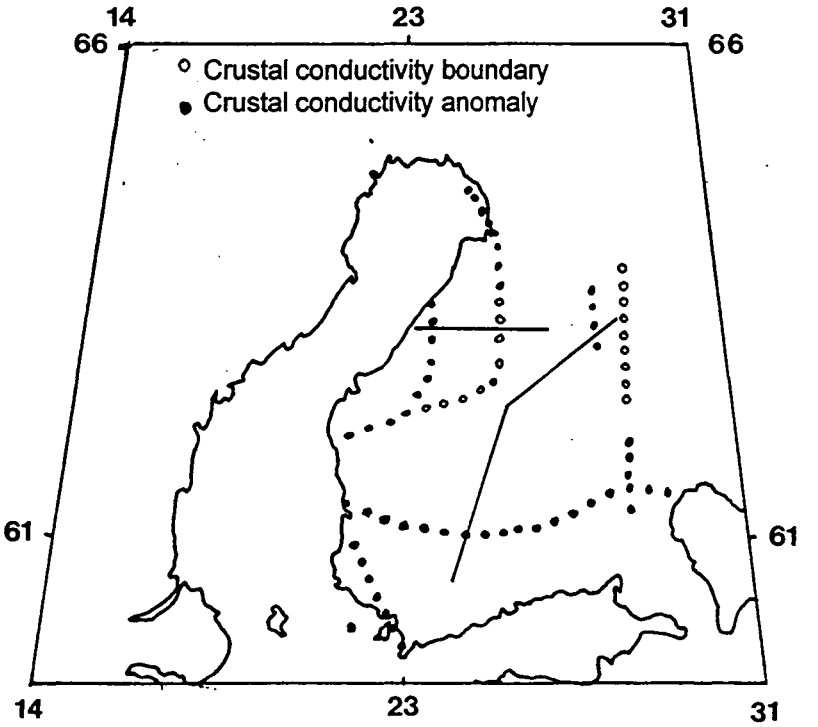


Fig. 3.9 Division of crust into blocks based on changes in conductivity, with the magnetotelluric profiles in Fig. 3.7 shown (after Pajunpää et al., 1987)

4. Field Work : Acquisition of the Data

4.1 Introduction: Aims of the Experiment

A number of questions and problems arise from the geological background and the results of previous geophysical investigations. Among these are: the exact location of the division between Archaean and Proterozoic crust; the relationship between the Skellefte Ore Zone and the Ladoga-Bothnian Bay Shear Zone (Fig. 2.1a); the presence of the Baltic-Bothnian Megashear; the cause of depressions in the Moho and the relationship between them; the cause of gravity and aeromagnetic anomalies in the Bothnian Sea; and the tectonic significance of bathymetric features such as the Aranda Rift and Strömmingsbåden Scarp (Fig. 2.9)

The BABEL project in the Gulf of Bothnia was designed with many of these questions in mind (Fig. 1.1). The Bothnian Bay experiment (BABEL Lines 2, 3 and 4) concentrated on the Archaean - Proterozoic boundary and the Skellefte Zone - Ladoga-Bothnian Bay Zone relationship.

In the Bothnian Sea, Line 1 was positioned to intersect the large gravity anomaly in the north (Figs. 3.5, 4.1). By running the line longitudinally along the axis of the Sea, a great enough length (>300 km) could be achieved to obtain velocity information from the lower crust and upper mantle at wide angle. This should allow any Moho depression running NE across the Sea to be imaged. The line also passes directly over the Strömmingsbåden Scarp. Line 6 runs parallel, to the west, so that rays shot into stations on the Finnish coast from Line 6 would have their midpoints beneath Line 1, and vice versa.

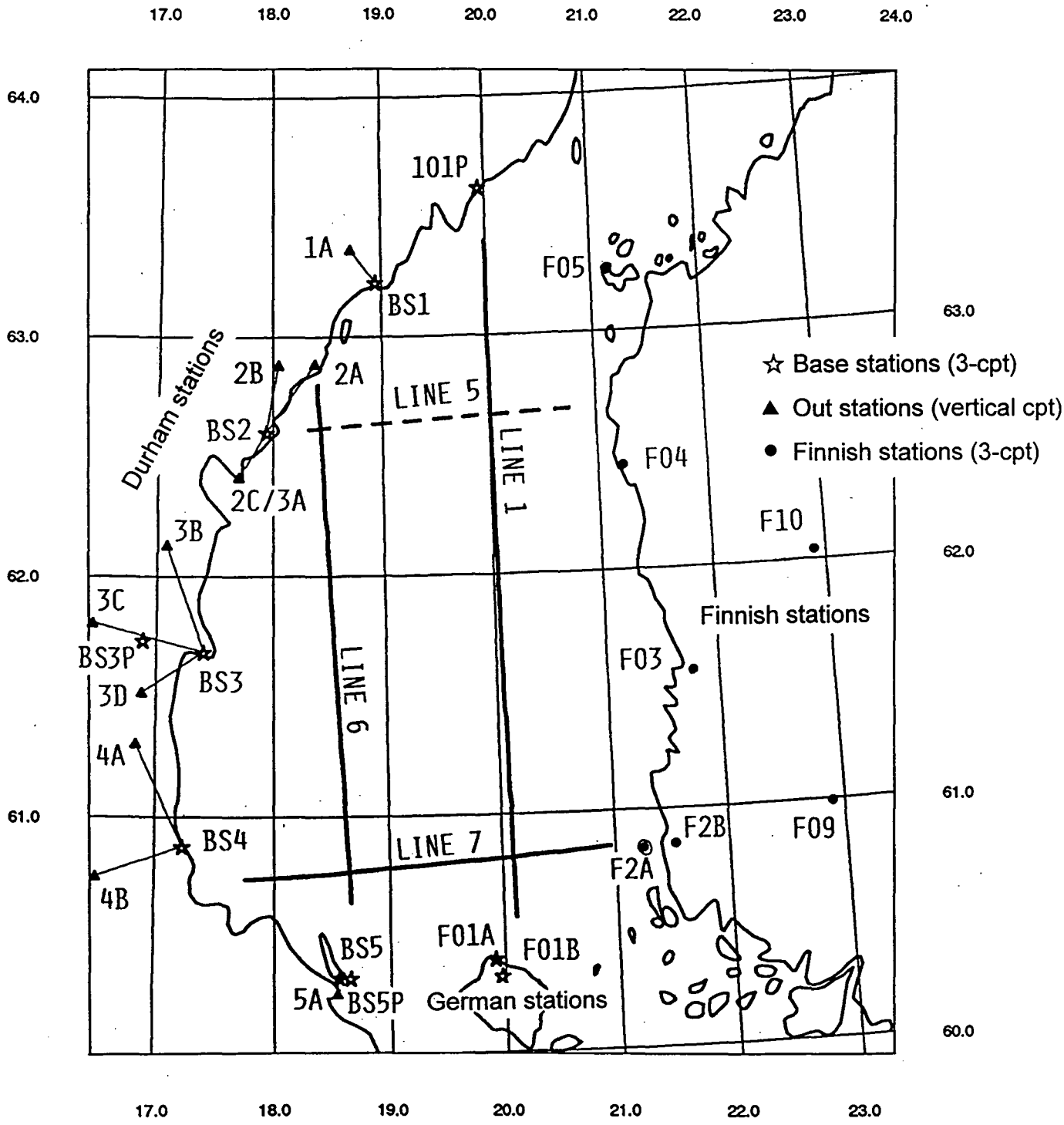


Fig. 4.1 Shots and recording stations of the BABEL project in the Sea of Bothnia

The wide angle recording stations were distributed all around the coastline of the Bothnian Sea to give a large set of intersecting raypaths, effectively giving three dimensional coverage of the region (Fig. 4.1). The station - shot separations were calculated so as to target mainly the mid and lower crust, and the Moho. To achieve a large enough offset, some stations were located well inland.

Two perpendicular lines (5 and 7) were planned at the northern and southern ends of the Sea in order to tie in the north-south profiles, and also to look for evidence of the Baltic-Bothnian Megashear. Line 7 had the additional aim of identifying the cause of the magnetic anomaly northwest of Åland. Unfortunately, due to limited funds, Line 5 could not be shot.

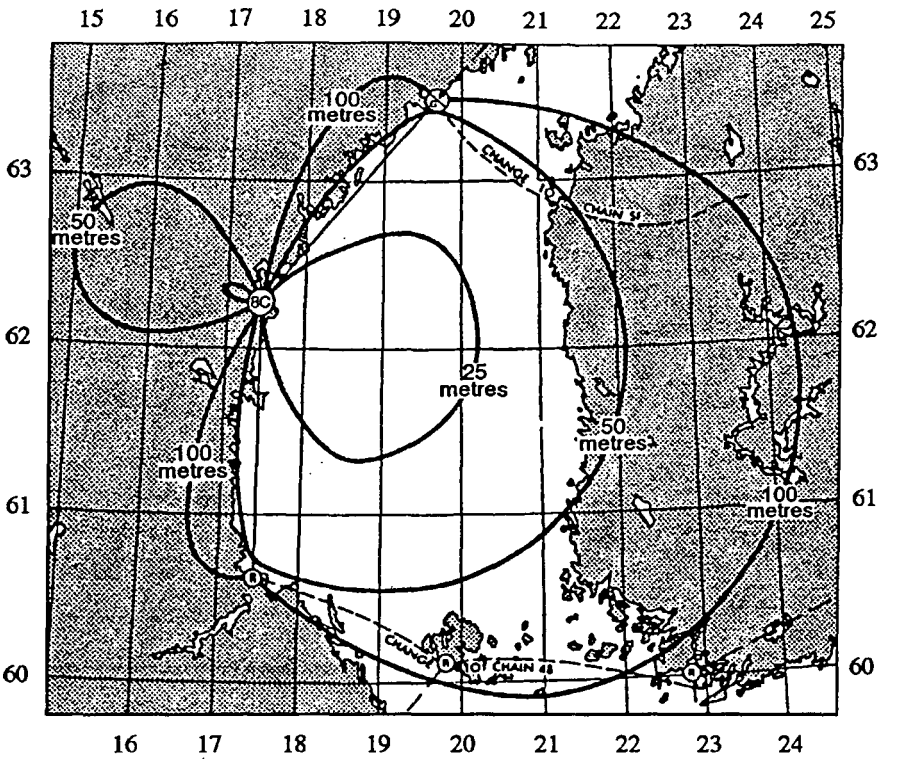
4.2 Marine work

The marine work was carried out by the Prakla-Seismos ship S. V. Mintrop.

4.2.1 Ship Navigation

The navigation system used aboard ship was the DECCA system. Fig. 4.2 shows the full daylight coverage and 68% fix repeatability for the Southern Bothnian chain (8C). Table 4.1 gives the random fixing errors for the accuracy contours in Fig. 4.2b. The appropriate light conditions for Line 1 are found in Fig. 4.3. Where possible, two DECCA chains were used, Doppler sonar was continuously recorded and all possible Transit satellite fixes were taken. This decreased the shot location errors considerably and

a)



b)

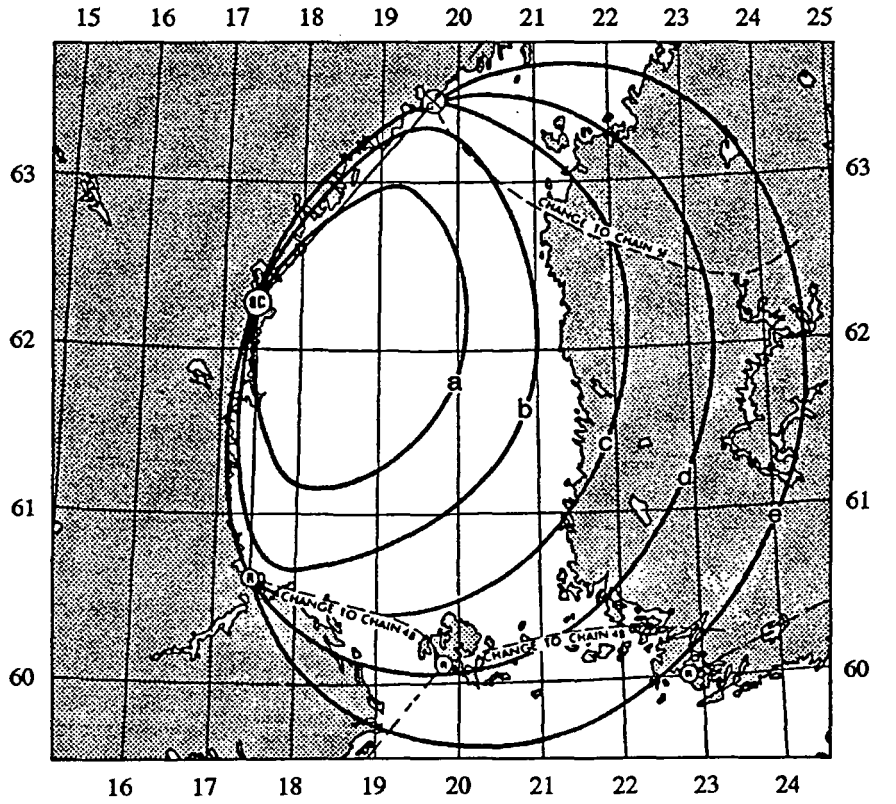


Fig. 4.2 DECCA navigation system in the Sea of Bothnia:

a) full daylight coverage

b) 68% fix repeatability (see also Table 4.1 and Fig. 4.3)

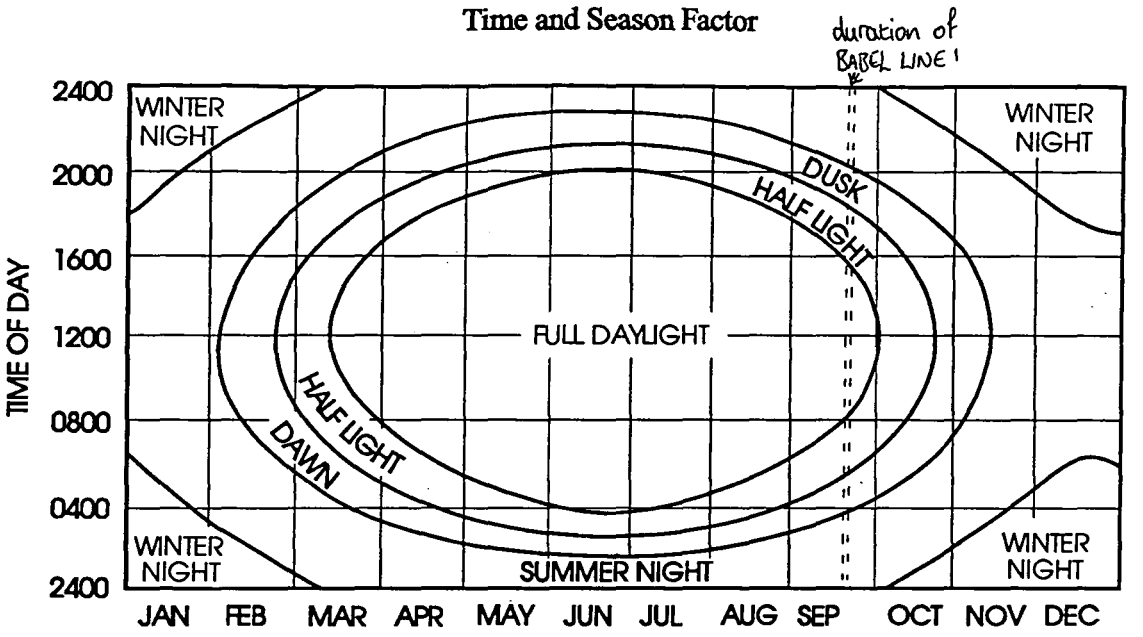


Fig. 4.3 Daily and annual variation in light conditions in the Sea of Bothnia

Random fixing errors at sea level (m) : 68% probability level

DECCA period See Time and Season Factor diagram	Contour				
	a	b	c	d	e
Half light	<200	<200	<200	250	450
Dawn / Dusk	<200	<200	250	450	950
Summer night	<200	250	450	950	1850
Winter night	200	350	690	1400	2800

Table 4.1 DECCA navigation system random fixing errors for the accuracy contours in Fig. 4.2b (DECCA manual)

the relative position of shots can be considered accurate to within a few metres, with a constant error in the line position of up to 200 m.

Line 1 was shot in three separate segments, the start and end points of which are given in Table 4.2. These are shown diagrammatically in Fig. 4.4.

4.2.2 Source specifications

The source used for the profiles was the Prakla-Seismos "F120/090/7.5" array of 42 airguns arranged in 6 strings of 8 as shown in Fig. 4.5. The total size of the array was 120.6 l (7359 cu. in.) and it was towed at a depth of 7.5 m. The source signature recorded by a hydrophone in the far field and its frequency spectrum are given in Fig. 4.6. The airguns were fired every 25½ seconds giving a nominal shot spacing of 75 m.

4.2.3 Hydrophones

Normal incidence data was collected with a 3 km long hydrophone streamer towed at a depth of 15 m. The hydrophones were divided into groups of sixtyfour with a group spacing of 50 m. 25 seconds of data were recorded for each shot with a sample interval of 4 ms. A low cut frequency filter was applied at 3 Hz and a high cut at 102 Hz.

Line		shot point	latitude	longitude	date	time
1	start	101	60° 33' 59.04"	20° 06' 05.00"	21-Sep	19:20:37.98
	end	2710	61° 57' 32.09"	20° 02' 54.11"	22-Sep	13:34:16.86
1A	start	3148	61° 56' 47.74"	20° 02' 56.10"	22-Sep	17:08:03.48
	end	4507	62° 51' 36.67"	20° 00' 47.23"	23-Sep	05:01:33.23
1B	start	5454	62° 49' 35.05"	20° 00' 51.20"	23-Sep	07:12:27.99
	end	6306	63° 23' 59.19"	19° 59' 59.87"	23-Sep	14:34:20.39

Table 4.2 Line coordinates for the three segments of BABEL Line 1

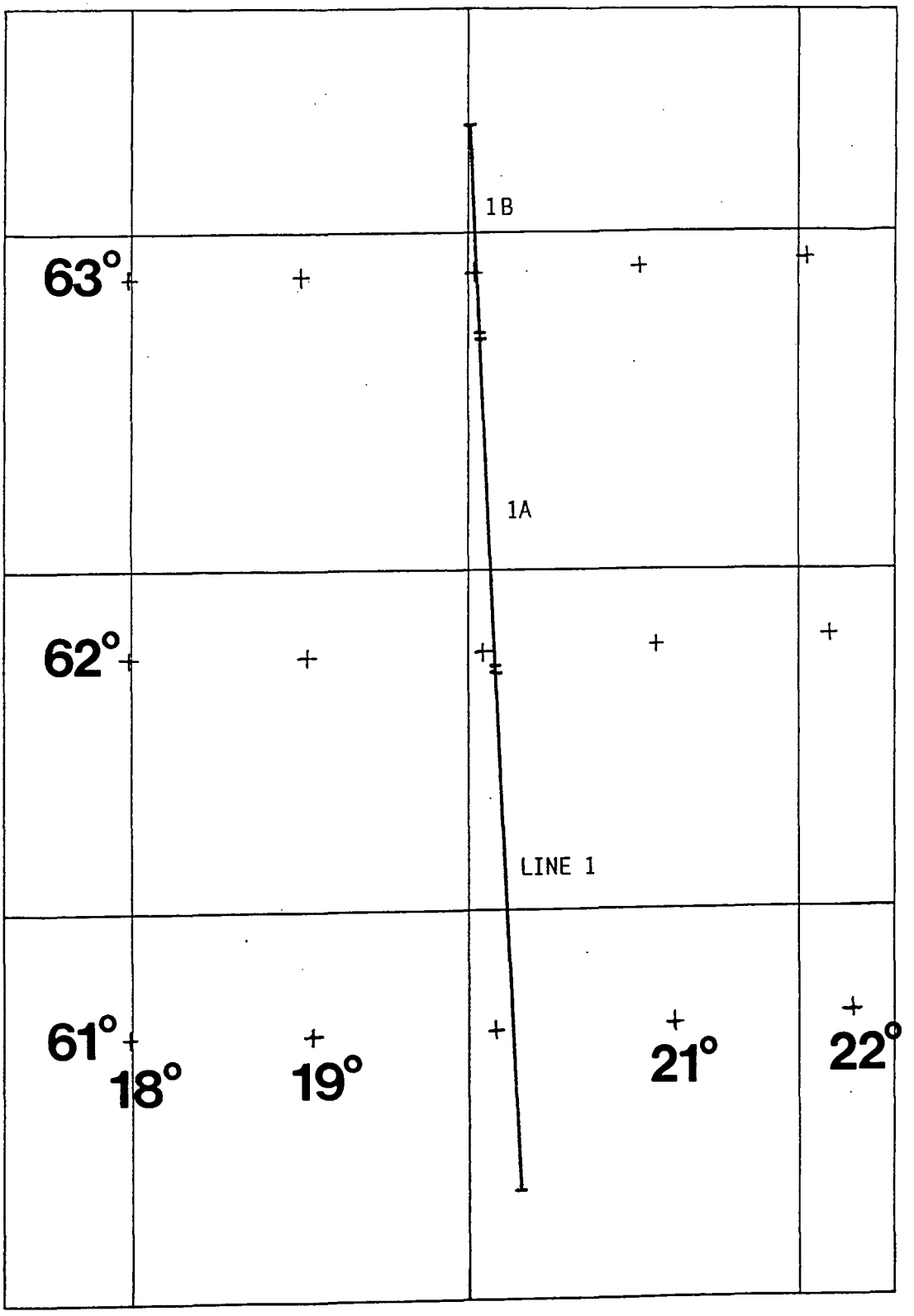


Fig. 4.4 Line 1 coordinates

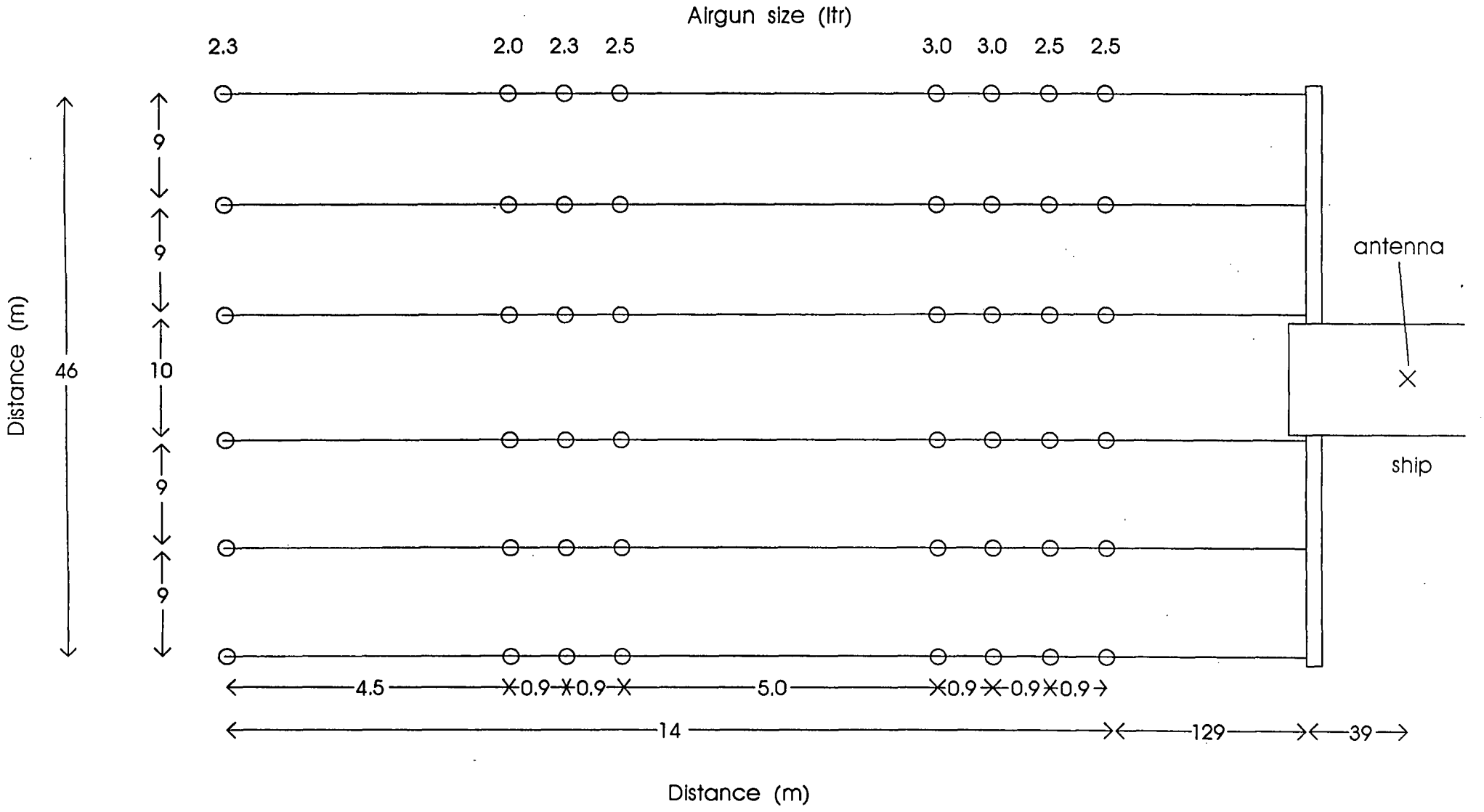
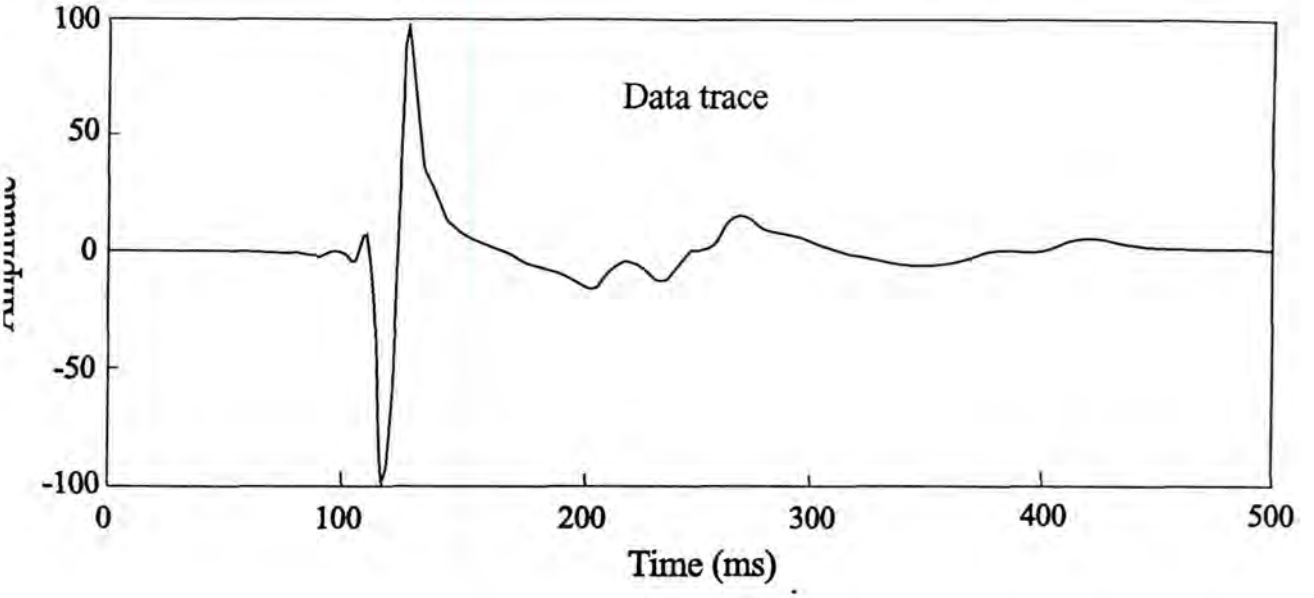


Fig. 4.5 Prakla-Seismos F120/090/7.5 airgun array



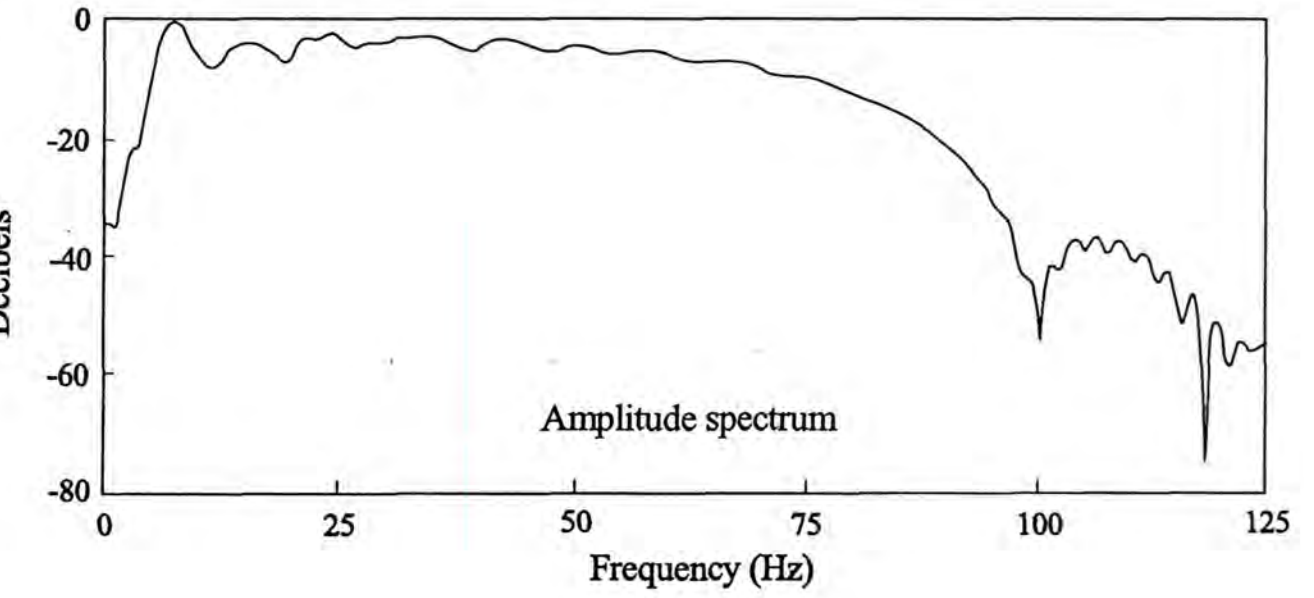


Fig. 4.6 Far field source signature and frequency spectrum recorded by hydrophone.

Source array depth = 6.0 m

Hydrophone depth below source = 108 m

4.3 Land stations

4.3.1 Station setup

Durham's part in the fieldwork involved installing and operating the wide angle seismic stations along the Swedish coast of the Sea of Bothnia (Fig. 4.1).

A system of up to four outstations linked by radio telemetry to a central recording station was employed to increase the coverage. Five such base stations (BS) were deployed in the low lying coastal regions while another ten outstations were set further inland at greater elevations to give line of sight between base station and outstation. Three component data were acquired at each base station whilst only the vertical component was measured at the outstations.

Each outstation consisted of a vertical seismometer, an amplifier modulator, a radio transmitter, an aerial and a 12-volt battery power supply. Most of the seismometers were positioned directly on bedrock which was found commonly outcropping in Sweden. In a few cases, a shallow pit had to be dug to reach the bedrock. At one station (2A, Fig. 4.1), solid bedrock could not be found and the seismometer was sited on loose, crumbly, weathered sedimentary rock. The amplifier modulator was placed near the seismometer and both were protected from the elements either with an upturned plastic bucket or with a wide piece of plastic piping covered by a sheet of plywood. The transmitter was strapped, inside a waterproof bag, to the aerial mast which was erected nearby.

The base stations required more hospitable conditions than the outstations and were therefore sited in various sheds and houses, or in the

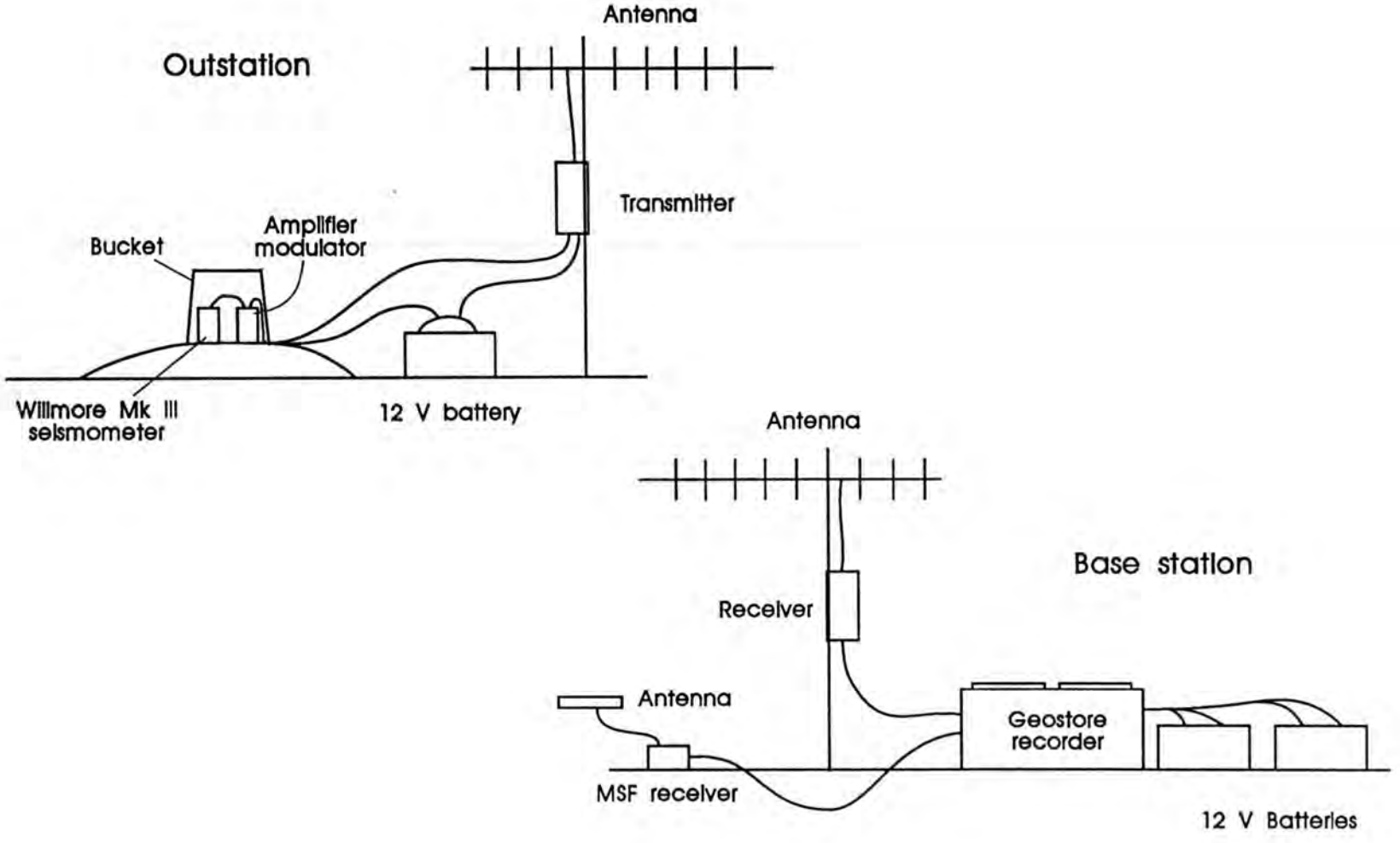


Fig. 4.7 Wide angle recording outstation - base station arrangement

case of BS3, in a tent. Each consisted of three seismometers (one vertical and two horizontal), a Geostore recorder, aerials and receivers, and an MSF radio time signal receiver. Power was supplied by two 12-volt batteries. The seismometers were installed as for the outstations. A schematic diagram of the outstation - base station arrangement is shown in Fig 4.7.

All the stations used Willmore Mk III variable period seismometers. These generate an E.M.F proportional to the ground velocity. The seismometers dominate the low frequency response characteristics of the acquisition system. This response is controlled by the resonant frequency and the damping. The frequency response curve is optimally flat at a damping of h = 0.7, where h is the ratio of damping to the critical damping. Fig. 4.8 shows the frequency response for the Willmore Mk III at three values of h. For BABEL, the seismometers had resonant frequencies of about 1.5 Hz and the damping was set at h ~ 0.7.

The amplifier modulator amplifies the seismometer output and converts it to an F.M. signal. The gain was set at 7 for all seismometers except those at BS5 where it was increased to 8. The UHF radio links between outstation and base station operated at frequencies ranging between 439.7 and 440 Mhz with a bandwidth of 25 kHz.

In addition to the Geostore stations, three component data were recorded digitally with two Geotech PDAS (Portable Data Acquisition System) instruments. One of these (BS3P) was located near BS3 for the duration of the experiment while the second was moved to record in-line data from both the northern end of Line 1 (station 101P) and the southern end of Line 6 (station BS5P).

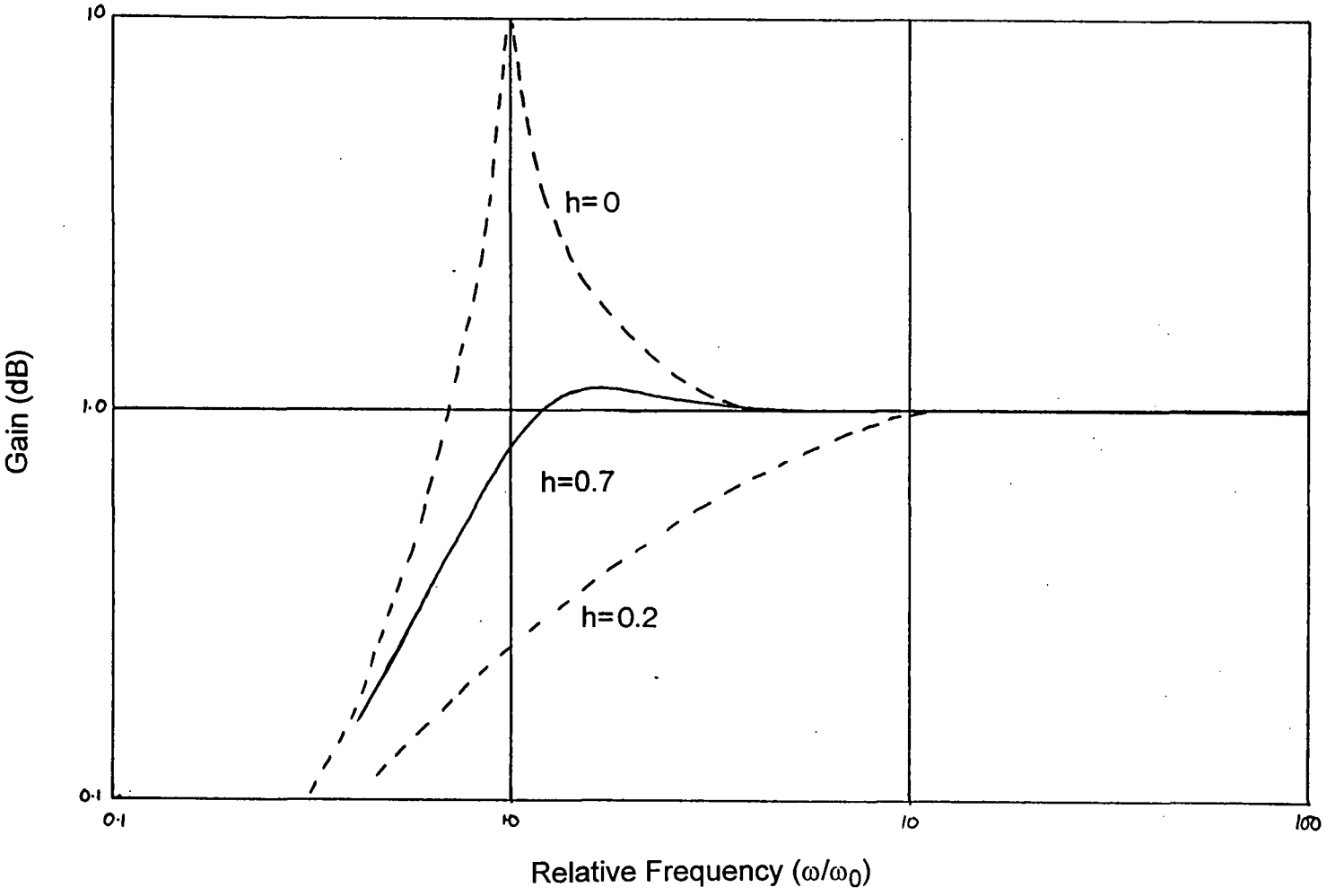


Fig. 4.8 Frequency response of the Willmore Mk III seismometer (resonant frequency $\omega_0 \approx 1.5$ Hz)

Station	Equipment	Component	Latitude	Longitude	Elevation (m)	Comments
101P	PDAS	x, y, z	63° 36' 45.6"	19° 58' 23.4"	0	In garage of a house
BS1	Geostore	x, y, z	63° 13' 29.4"	18° 57' 39.6"	35	Stubbsand, in cellar of a house
1A	Geostore	z	63° 22'28.2"	18° 42' 03.0"	180	Nr Klingre
BS2	Geostore	x, y, z	62° 36' 31.2"	17° 57' 55.8"	130	Härnön, in amateur radio shed
2A	Geostore	z	62° 52' 55.8"	18° 21' 42.6"	115	Kåstaänget, "shaky hill"
2B	Geostore	z	62° 52' 37.8"	18° 02' 00.0"	135	Nr Milstabbana
2C/3A	Geostore	z	62° 24' 51.0"	17° 41' 33.6"	50	Nr Sörsidan, military training area
BS3P	PDAS	x, y, z	61° 44' 38.4"	16° 55' 27.0"	40	Nr Forsa, in house
BS3	Geostore	x, y, z	61° 41' 17.4"	17° 23' 48.6"	65	Nr Arnöviken, in a tent
3B	Geostore	z	62° 08' 16.2"	17° 05' 12.0"	330	Västansjökullen, military observation tower
3C	Geostore	z	61° 49' 04.8"	16° 24' 34.2"	245	Klotjärnsasen
3D	Geostore	z	61° 31' 11.4"	16° 53' 28.8"	245	Nr Storäsen
BS4	Geostore	x, y, z	60° 52' 25.2"	17° 17' 27.6"	20	Iggön, in farm shed
4A	Geostore	z	61° 12' 35.4"	16° 50' 19.8"	200	Nr Gullberg
4B	Geostore	z	60° 45' 09.6"	16° 30' 00.0"	290	Kungsberget
BS5P	PDAS	x, y, z	60° 18' 37.8"	18° 32' 16.8"	5	Rävsten
BS5	Geostore	x, y, z	60° 18' 42.0"	18° 36' 37.8"	10	Gräsö, old blacksmith's workshop
5A	Geostore	z	60° 15' 04.8"	18° 35' 55.2"	10	Nr Sundsveden

Table 4.3 Site locations of wide angle recording stations operated by Durham in Sweden

In total, these formed an array of 18 stations spanning a distance of over 500 km along the Swedish coast of the Sea of Bothnia. A summary of the site locations and station details is contained in Table 4.3. The coordinates are estimated to be accurate to ±100 m. This relatively high error estimate is due to the lack of significant landmarks on which to take compass bearings and the prevalence of dense forestation at many sites, obscuring the view.

4.3.2 Geostore recorder

The Geostore is an F.M. analogue recorder with 14 channels. Two of these are flutter tracks and a third records internally generated VELA time code. One of the remaining channels (channel 14) was used to record the MSF radio time signal from the Rugby transmitter, the reception of which varied from site to site and also with time of day. The data was recorded to ½ inch magnetic tape at a speed of 15/160 inches per second. This gave a bandwidth of 31.2 Hz whilst allowing a recording time of between two and three days per tape, essential for continuous recording given the distance between base stations and the time needed to travel between them. As an analogue system, no anti-alias filter was applied during acquisition or playback on a Store-14 tape deck. However, the signal was low pass filtered on demodulation to give a bandwidth dependent on the tape speed during recording. At 15/160 inches per second, the 3dB point was 31.2 Hz with a roll-off of 20 dB/octave (Fig. 4.9).

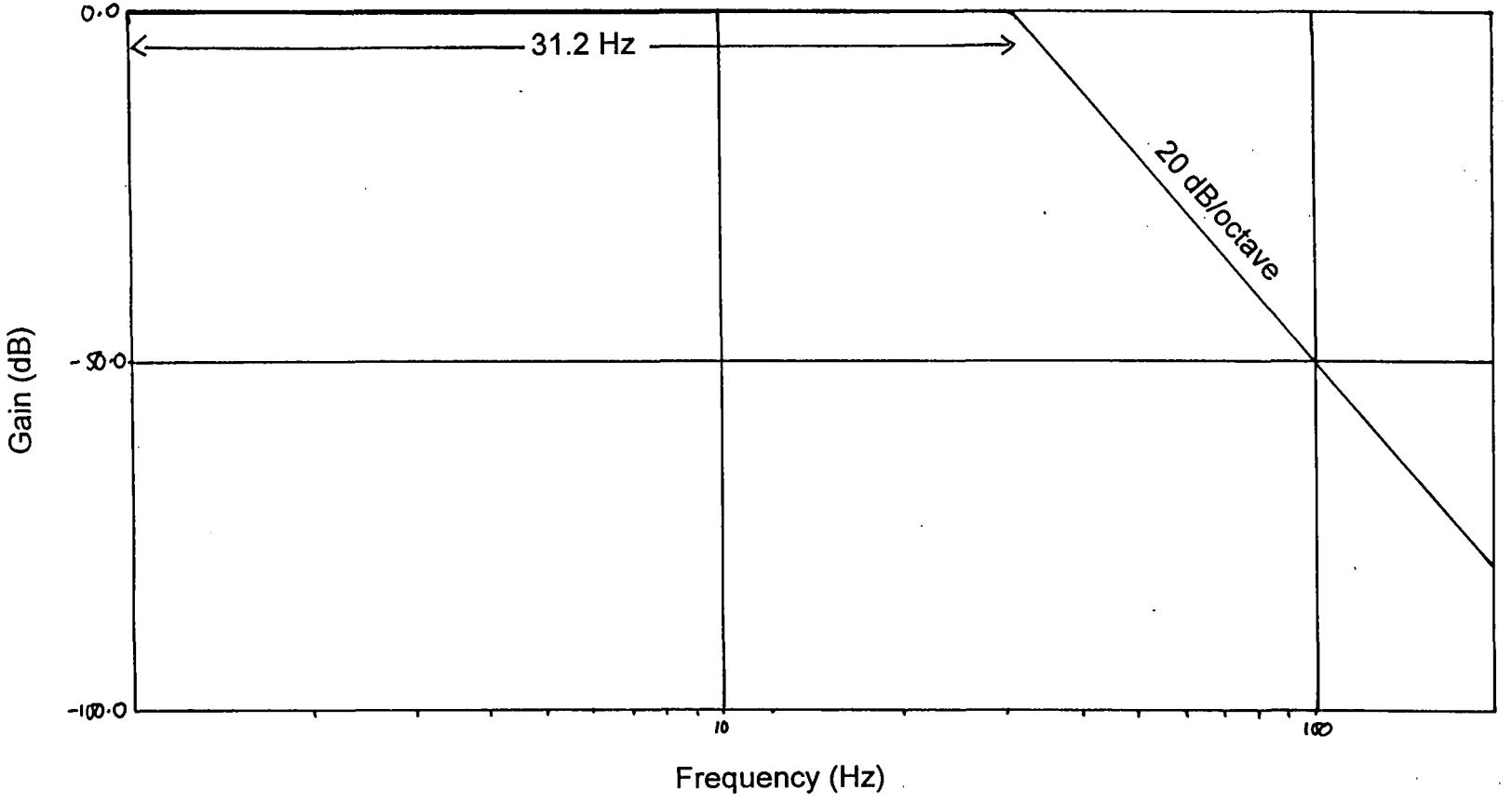


Fig. 4.9 Frequency response of the Store14 analogue demodulating filter.

4.3.3 PDAS recorder

The PDAS used a sample interval of 10 ms, chosen to give a bandwidth of 40 Hz whilst permitting a reasonable continuous recording length before the memory was full. Two anti-alias filter were applied to the data: a 6 pole (36 dB/octave) Butterworth analogue filter with a bandwidth of 200 Hz, and a sample rate - dependent digital filter. For the chosen sample rate, this had a flat frequency response up to 40 Hz with 372 dB/octave anti-aliasing filtering (Fig. 4.10). To obtain the maximum dynamic range for minimum memory consumption, data was stored in gainranged integer format, i.e. as 2 byte integers with a 14 bit mantissa and a 2 bit gain code. Since the PDAS only had a small memory capacity (1.6 Mbytes), the data had to be off-loaded to the hard disk of an IBM compatible laptop computer (Zenith or Toshiba) every 20 minutes. This operation took approximately 3 minutes each time during which no data could be recorded. The data on the hard disk were dumped to tape once an hour with no interruption to acquisition. Due to a technical hitch in the link between the PDAS, laptop computer and tape drive, no data were acquired from the southern third of Line 1 by the instrument at station 101P. PDAS internal clock time was written into the header at the start of each 20 minute data file. The MSF time signal was also recorded at each PDAS station.

4.3.4 Other stations

The stations at the southern end of Line 1 (F01A,B) on the island of Åland were operated by a group from the GEOMAR Institute in Kiel,

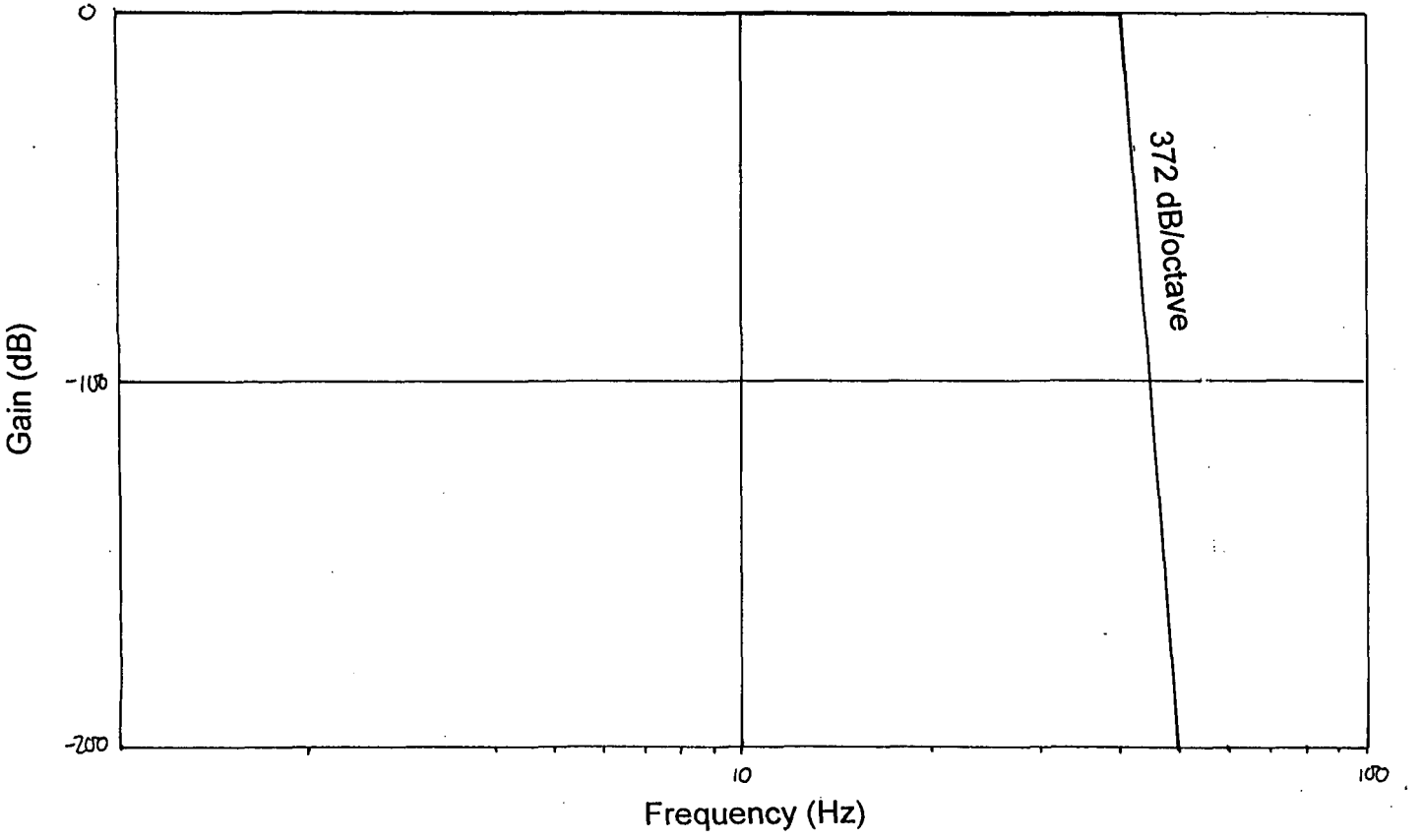


Fig. 4.10 Frequency response of the PDAS digital anti-alias filter.

Germany. Two types of geophones were used here: a three component instrument with a low frequency cut off of 2 Hz and a single, vertical component geophone, low cut 4.5 Hz. The vertical component geophones were grouped into linear arrays of six and crossed arrays of twelve, each geophone being cemented to the ground with gypsum plaster to improve coupling. Two 3-component geophones, five chains of six and one crossed array of twelve geophones were located around each of two sites (F01A and F01B) on the island (Table 4.4). An estimate of the error in the site coordinates is not available so the estimate for the Durham stations is assumed. The data were recorded with a sample interval of 5 ms using 4 and 8 channel PCM 5800 instruments.

Groups from Helsinki and Oulu Universities operated the stations in Finland, also using digital instruments. Table 4.5 shows the site locations and again the accuracy of these is not known. These data were not available in digital form for use in this study, although paper record sections were obtained.

Station	Equipment	Component	Latitude	Longitude	Elevation (m)	Comments
F01A 114/1,3	PCM 5800	x, y, z	60° 23' 50.5"	19° 53' 51.0"	70	3-component Lippman geophone (2 Hz)
4		z	60° 23' 48.5"	19° 53' 55.0"	65	chain of 6 geophones (4.5 Hz), linearly spaced
F01A 118/1,3	PCM 5800	x, y, z	60° 23' 51.0"	19° 53' 52.0"	70	3-component Lippman geophone (2 Hz)
4		z	60° 23' 50.5"	19° 53' 40.0"	75	2 chains of 6 geophones (4.5 Hz), crossed
5		z	60° 23' 49.5"	19° 53' 27.0"	85	chain of 6 geophones (4.5 Hz), linearly spaced
6		z	60° 23' 57.5"	19° 53' 35.0"	85	chain of 6 geophones (4.5 Hz), linearly spaced
7		z	60° 23' 58.5"	19° 53' 59.0"	55	chain of 6 geophones (4.5 Hz), linearly spaced
8		z	60° 23'54.0"	19° 53' 57.0"	60	chain of 6 geophones (4.5 Hz), linearly spaced
F01B 124/1,3	PCM 5800	x, y, z	60° 19' 26.0"	19° 58' 26.0"	45	3-component Lippman geophone (2 Hz)
4		z	60° 19' 22.5"	19° 58' 26.0"	55	chain of 6 geophones (4.5 Hz), linearly spaced
F01B 128/1,3	PCM 5800	x, y, z	60° 19' 27.0"	19° 58' 16.0"	45	3-component Lippman geophone (2 Hz)
4		z	60° 19' 29.0"	19° 58' 29.0"	40	2 chains of 6 geophones (4.5 Hz), crossed
5		z	60° 19' 26.5"	19° 58' 08.0"	65	chain of 6 geophones (4.5 Hz), linearly spaced
6		z	60° 19' 21.0"	19° 58' 05.0"	60	chain of 6 geophones (4.5 Hz), linearly spaced
7		z	60° 19' 02.0"	19° 58' 05.0"	65	chain of 6 geophones (4.5 Hz), linearly spaced
8		z	60° 19' 39.5"	19° 58' 10.0"	55	chain of 6 geophones (4.5 Hz), linearly spaced

Table 4.4 Site locations of wide angle recording stations operated by Kiel in Åland

Station	Latitude	Longitude	Elevation (m)
F2A	60° 50' 52.1"	21° 14' 48.6"	2
F2B	60° 51' 22.8"	21° 31' 29.0"	17
F03	61° 39' 03.2"	21° 35' 31.5"	5
F04	62° 26' 30.9"	21° 13' 32.2"	0
F05	63° 18' 52.4"	21° 10' 11.1"	5
F09	60° 59' 45.2"	22° 53' 47.8"	92
F10	62° 02' 44.2"	22° 53' 44.3"	165
S13	56° 07' 01.6"	15° 50' 15.0"	0
S04	55° 30' 01.3"	14° 14' 07.8"	0

Table 4.5 Site locations of wide angle recording stations operated by Helsinki and Oulu in Finland

5. Processing

5.1 Introduction

Data collected by the Durham group using the PDAS and Geostore equipment were processed entirely at Durham. Since very little software was available to us at the start of the project, most of the processing programs were written by the author, with some also by P. A. Matthews (Appendices 1-4). This software development comprised a major part of the project. The processing sequence is show in Fig. 5.1.

The data from station F01A which were acquired by the Geomar Institute, Kiel, were converted to SEGY format before being sent to Durham. The normal incidence data were processed initially by the contractors (Prakla Seismos) with further processing carried out by BIRPS and the Bullard Laboratories, Cambridge.

5.2 Wide Angle Processing

5.2.1 PDAS data : conversion to internal trace format (Stage 1)

Since the PDAS data were already in a digital format, these were the first to be processed. The raw data consisted of a series of 20 minute long files on DC 2000 magnetic tape cartridges, one MSF time code and three data channels being recorded in separate file sequences. Each file header, an example of which is given in Table 5.1, contains the start-of-file time as measured by the internal clock.

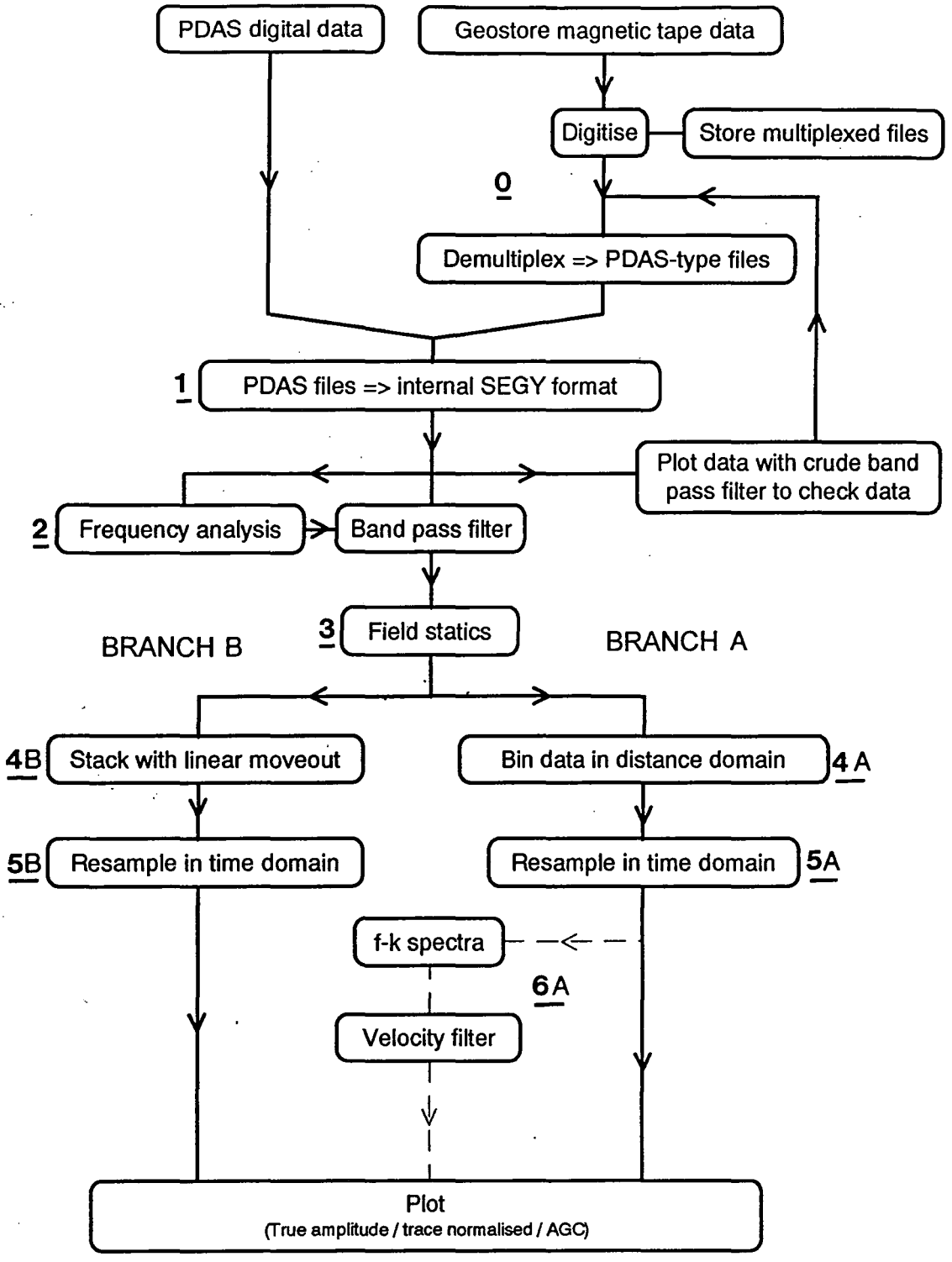


Fig. 5.1 Processing sequence for wide angle data

Table 5.1. a) PDAS file header specification

Header		Description
DATASET	P1107018270	File name
FILE_TYPE	LONG	Data type (integer=2 byte, long=4 byte)
VERSION	next	Set filename to next sequence number
SIGNAL	Z	Channel name
DATE	9-27-89	Date at start of file
TIME	13:22:56.50 (500)	Time at start of file
INTERVAL	0.010	Sample interval (seconds)
VERT_UNITS	Counts	Always counts
HORIZ_UNITS	Sec	Always seconds
COMMENT	GAINRANGED	Data type
DATA		Beginning of binary data

b) Breakdown of PDAS file name

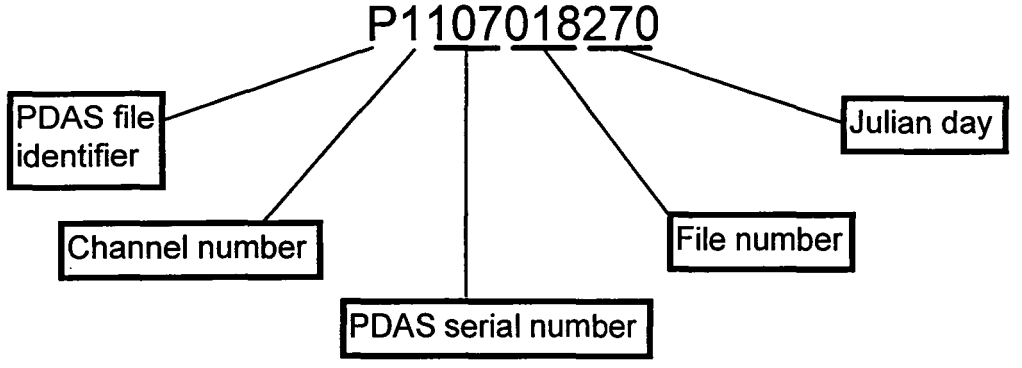


Table 5.2 Internal SEGY specification

a) ASCII file header (3200 bytes) - project name, processing details, etc.

b) Binary file header (4 byte integers) - 400 bytes

1- 4	Number of traces per record
5- 8	Number of non-seismic traces per record
9- 12	Not used
13- 16	Sample interval (s)
17- 20	Not used
21- 24	Number of samples per data trace
25- 52	Not used
53- 56	Data format : 1=floating point (4 byte) 2=fixed point (4 byte) 3=fixed point (2 byte) 4=fixed point w/gain code (4byte)
57-200	Not used
201-204	Reel number for this line
205-208	Line number
209-212	Measurement system : 1=metres, 2=feet
213-400	Not used

Table 5.2 (continued)

c) Binary trace header - 512 bytes / record

13- 16	trace/ channel/ station number
17- 20	shot/ event number
29- 30	trace identification code: 1=seismic data 2=dead channel 3=dummy channel
31- 36	not used
37- 40	horizontal distance shot-receiver
41- 44	receiver elevation
45- 48	surface elevation at source
49- 52	source depth below surface
53- 56	datum elevation at receiver
57- 60	datum elevation at source
61- 64	water depth at source
65- 68	water depth at receiver
69- 70	scaler for 41-68 to give real values : positive - multiplier, negative - divisor
71- 72	scaler for 73-88 to give real values : positive - multiplier, negative - divisor
73- 76	source coordinate X (longitude, positive to east)
77- 80	source coordinate Y(latitude, positive to north)
81- 84	receiver coordinate X
85- 88	receiver coordinate Y
89- 90	coord units: 1 = length, 2 = sec arc
91-108	not used
109-110	delay between shot instant and start of trace (msec part)
111-114	not used
115-116	number of samples in this trace
117-118	sample interval for this trace (μs)
119-120	not used
121-122	instrument gain constant
123-140	not used
141-142	alias filter frequency
143-144	alias filter slope
145-148	not used
149-150	low cut frequency
151-152	not used
153-154	low cut slope
155-156	not used
157-158	year } time at start of trace
159-160	day
161-162	hour
163-164	minute
165-166	sec
167-168	time basis : 1=local, 2=UT
169-186	not used
187-188	msec - time at start of trace
189-190	not used
191-192	year } time of event
193-194	day
195-196	hour
197-198	minute
199-200	sec
201-202	msec
203-206	not used
207-208	delay between shot instant and start of trace (sec part)
209-210	not used
211-214	reduction velocity (km s^{-1})
215-216	reduced time at start of trace
217-220	not used
221-224	four character station code
225-228	four character channel code
229-512	not used

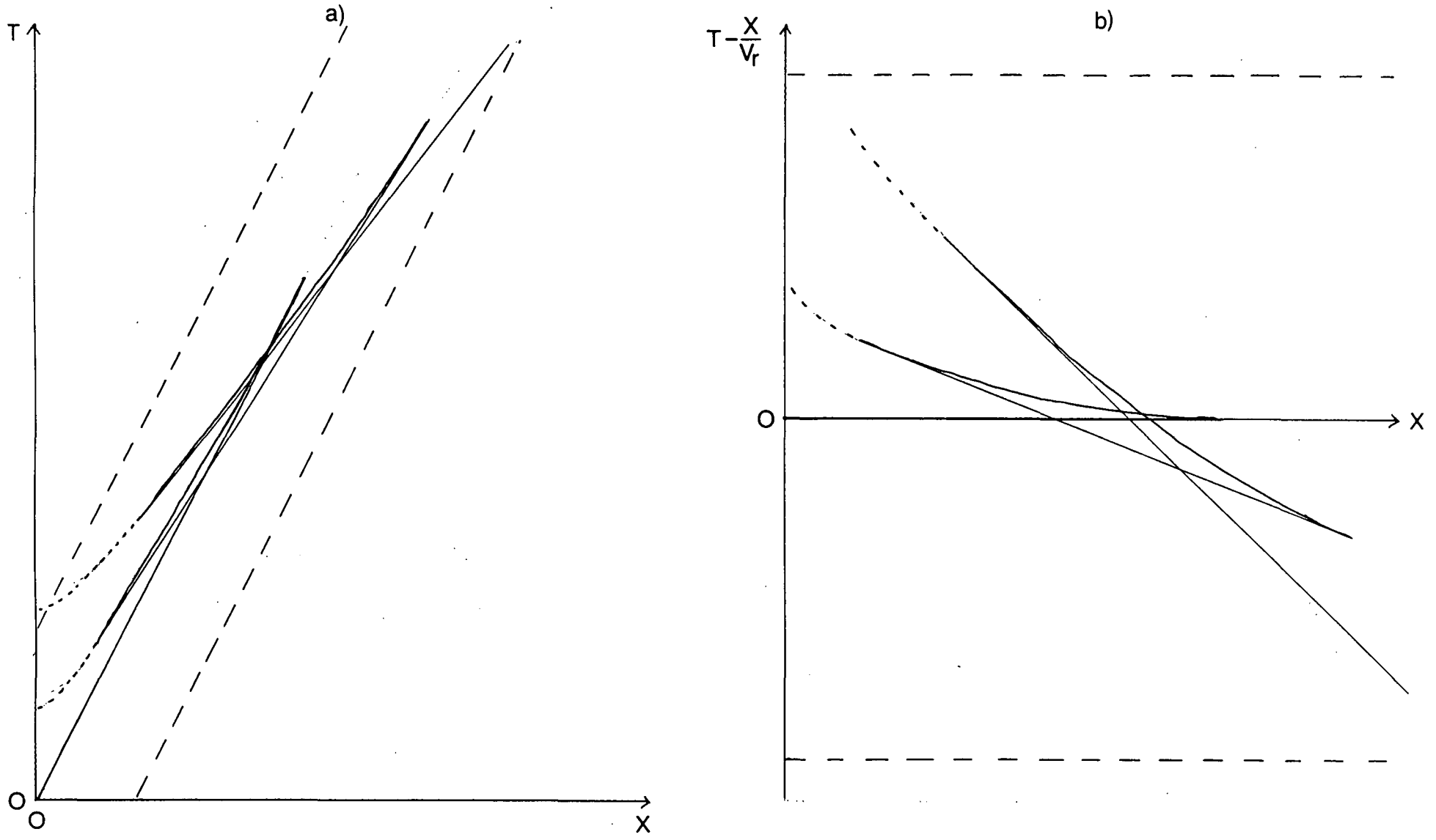


Fig. 5.2 Schematic representation of a wide angle seismic record

(a) before and (b) after reduction

This was checked against Universal Time (U.T.) by viewing the MSF time code channel and calculating the actual start-of-file time to the nearest sample (10 ms) to give a time correction value. Comparing values from several different points along the line showed there to be no measurable drift in the internal clock so the time correction could be applied as a constant offset.

After degainranging, i.e. converting the data from 2 byte 14/2 gainranged format to 4 byte integers, the data were put into an internal trace format using software developed by P. A. Matthews (1993) (CONVERT, Appendix 4). This format, based on the SEGY system, was devised by D. L. Stevenson of Durham University. It consists of an ASCII file header, a binary file header and two direct access binary files, the first containing one data trace per record and the second containing one trace header for each data trace. Table 5.2 shows the content of the binary file and trace headers. This format will hereafter be referred to as ISEGY.

Data were read in from successive PDAS files and, based on the start-of-file time, clock correction and shot instant, 40 second traces were written out. To fit all the useful data into 40 seconds for the entire distance range and cut out most of the unwanted information prior to the first arrival, a reduced form was used in which the trace start time t_s is given by :

$$t_s = t_e + \frac{X}{V_r} - t_l$$

where t_e = event time (shot instant), X = shot-receiver distance, V_r = reduction velocity (6 km s^{-1}) and t_l = trace lead-in time (6 seconds). The shot-receiver distance was calculated from the shot and receiver coordinates, both of which were entered in the trace headers.

The effect of this transformation is that a first arrival with apparent velocity equal to V_r will occur 6 seconds into the trace, reducing the required trace length.

Time from the start of the trace is thus the reduced travel time T_r rather than the actual travel time, where

$$T_r = T - \frac{X}{V_r}$$

Plotting reduced travel time against offset allows the expansion of the time axis to reveal more detail (Fig. 5.2).

The conversion to ISEGY format was carried out on an IBM-compatible DELL PC under MS-DOS but since all subsequent processing would be done on UNIX-based SUN workstations, the output binary trace and header files were written in SUN binary format rather than IBM. Although the hard disk capacity of the DELL was over 300 Mbytes, the version of MS-DOS used required that this be partitioned into nine separate 'drives' of 32 Mbytes each, and one of 16 Mbytes. A whole line would not fit on a drive at one time so several files were produced for one line and transferred to the SUN workstation separately. The files were then combined into one using the COMBINE9 program, written by the author (see Appendix 1). This program simply reads in data from each fileset in turn and writes it out sequentially to a single new fileset.

5.2.2 Digitisation of Geostore magnetic tape data (Preliminary stage 0)

The analogue data recorded on magnetic tape by the Geostore recorders were played out using a Store-14 tape player at 80 times the

recording speed. The data were digitised via the DT2821 data acquisition board and Global Lab software package (Data Translation, Inc.) on the DELL PC. These were controlled by a series of batch files and macros devised by Dr R. E. Long and extensively tested by the author.

Multiplexed digital data were acquired onto seven of the 32 Mbyte disk drives in turn. To keep the process reasonably fast and the data length acquired in one go large, only the internal time code and two data channels were read at one time.

Although no anti-alias filter was applied prior to digitisation, demodulation of the F.M. signal by the Store-14 playback system introduced a low pass filter with a bandwidth of 31.2 Hz and a slow 20dB/octave roll-off (see Chapter 4). A nominal sample interval of 5 ms was chosen in order to utilise this as an anti-alias filter, giving 45 dB of alias protection at the Nyquist frequency (100 Hz). At a sample interval of 5 ms, 12000 seconds of data from each channel could be stored on each disk drive so the whole of Line 1 fitted onto 13 such drives. The multiplexed data were copied to the Sun workstations for storage.

The data were demultiplexed into files similar to those produced by the PDAS equipment (not gainranged) so that they could be dealt with by the same software. The multiplexed internal time code data were read and automatically decoded with routines written by Dr R. E. Long so that a PDAS file could be written for every three minutes of data. The start of file time and mean sample interval over the length of the file was written into the header. Since the real time digitising frequency was constant, slight fluctuations in the tape speed caused variations in the data sample interval which was therefore not necessarily equal to the specified value of 5 ms. The mean sample interval for each 3 minute file was found by dividing 180

seconds by the number of samples in the file. Each drive was demultiplexed to the one free drive in turn and the ISEGY trace data written back over the multiplexed data on the original drive.

The internal and MSF time codes were compared independently at intervals along each tape reel by acquiring a short length of data from both channels and decoding the internal clock code at the same position as the first MSF minute marker. This gave an idea of the drift on the internal clock and a clock correction value for each tape reel.

Due to the variation in sampling rate over the whole dataset, the length in time of each trace produced after running through the PDAS_ISEGY software (Stage 1) was slightly different, although the number of samples per trace was necessarily constant.

After conversion to ISEGY format, the separate filesets for each channel (one from each drive of multiplexed data originally acquired from tape) were combined to give one fileset per channel per line using the program COMBINE9, as for the PDAS data. The data were plotted out with a crude boxcar band pass filter (4 - 25 Hz) to check for obvious timing errors (e.g. a step-out in the first arrival) and lost or corrupted data.

5.2.3 Frequency Analysis and Filtering (Stage 2)

5.2.3 a) Analysis

Firstly, any D.C. component in the data was removed using the program RMDC (see Appendix 1). The frequency content of a number of traces was then found by taking the amplitude spectrum of the Fourier transform (Fig. 5.3).

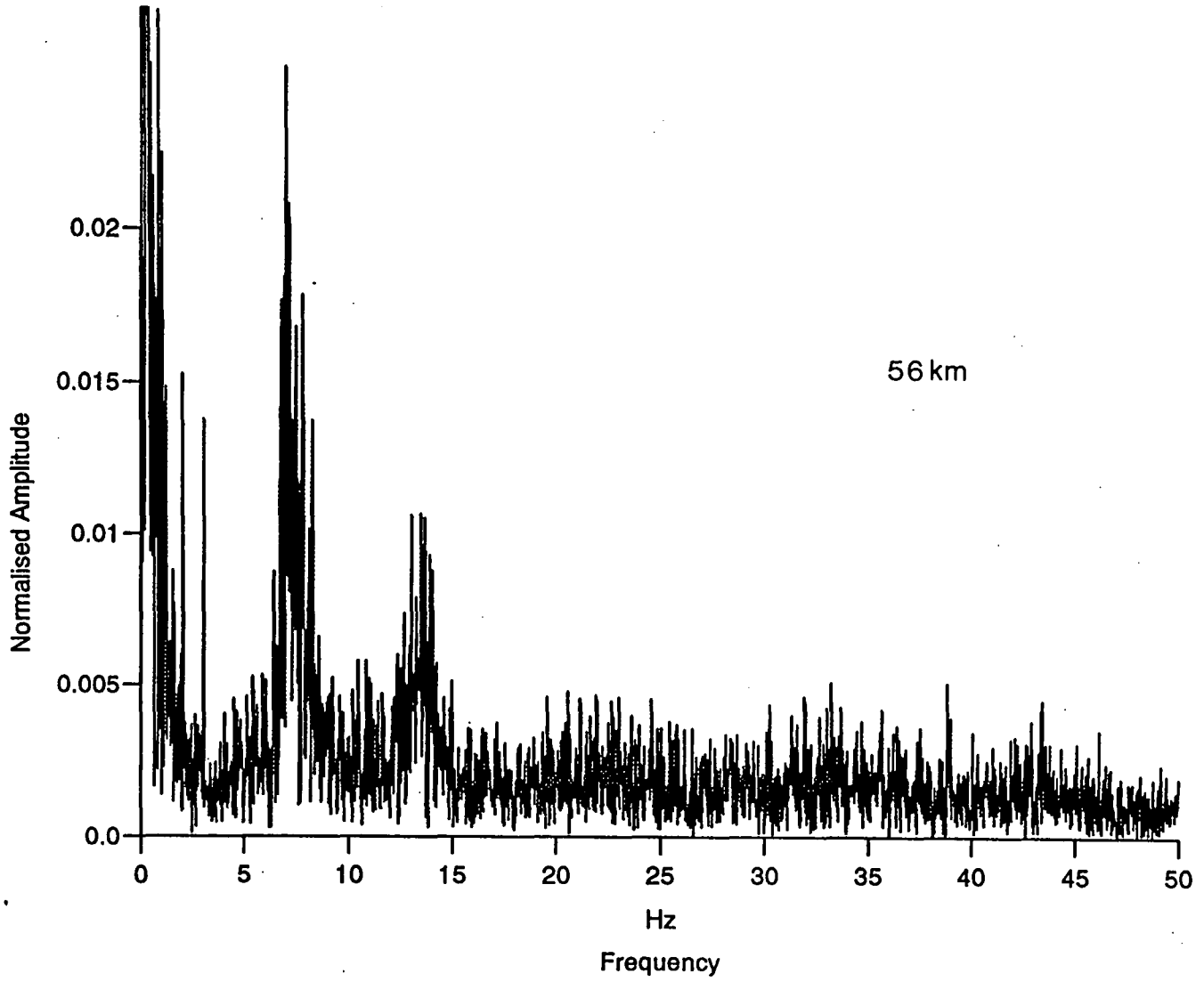


Fig. 5.3 Frequency amplitude spectrum of a single trace from station 1A.

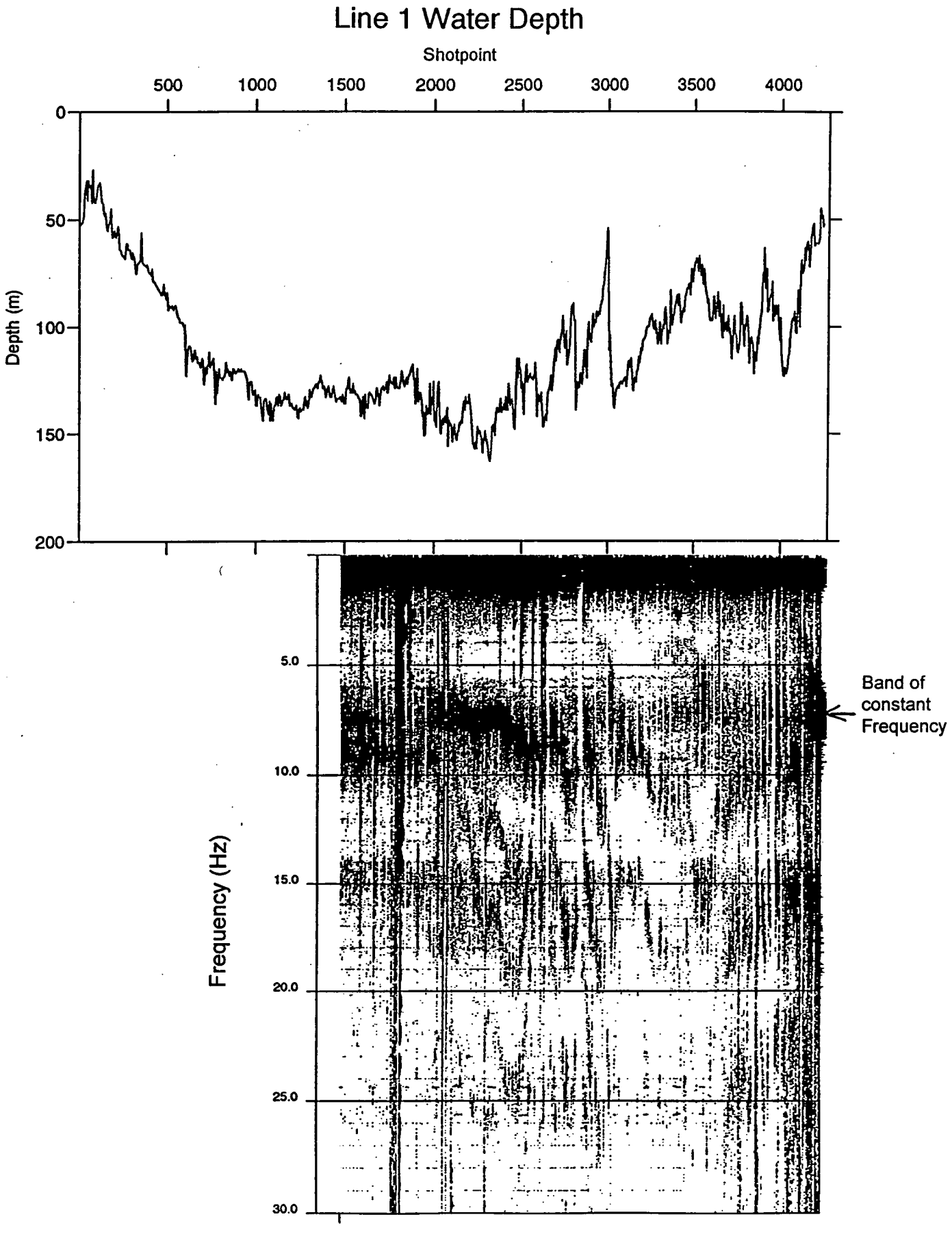


Fig. 5.4 a) Station 101P frequency amplitude spectrum vs. shotpoint

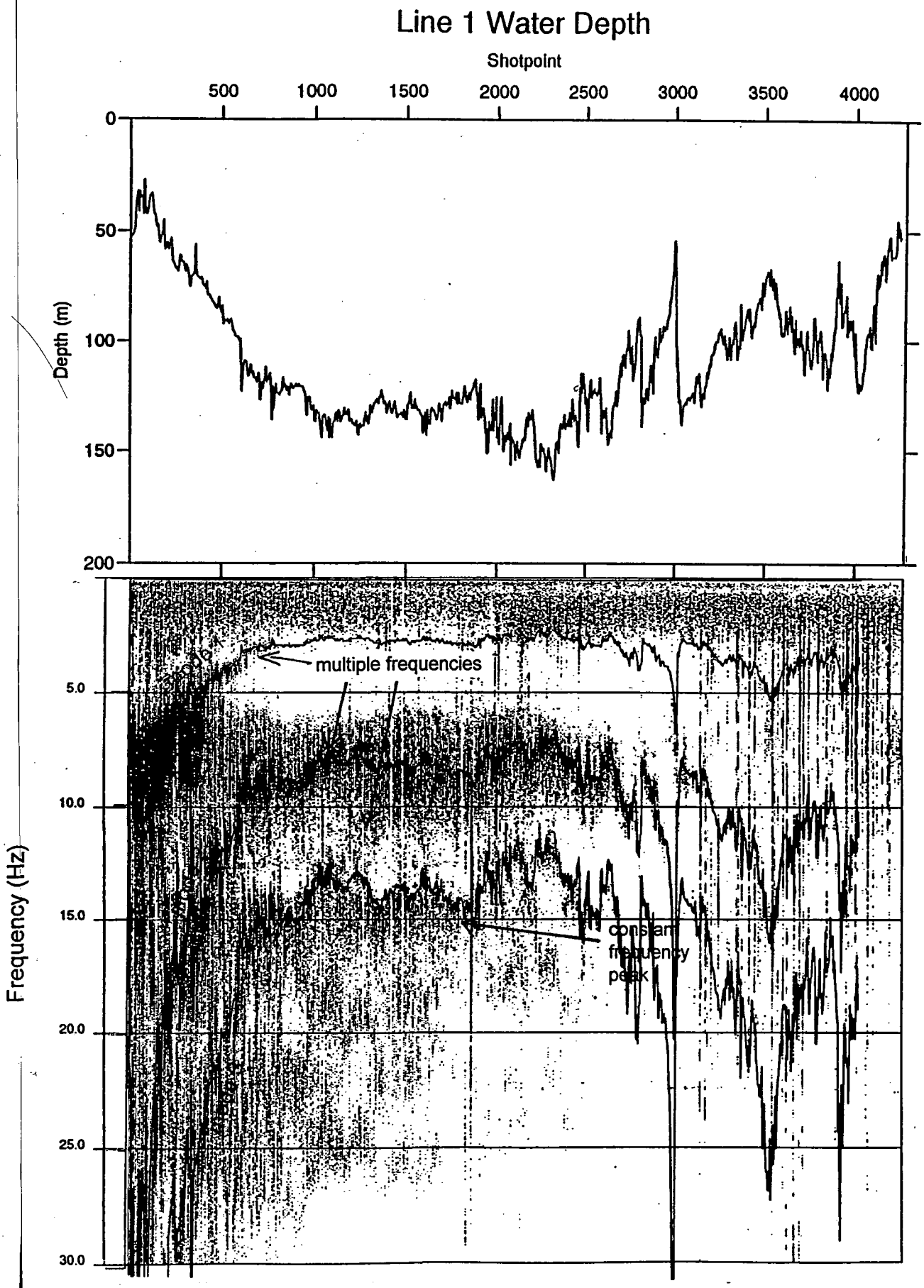


Fig. 5.4 b) Station F01A frequency amplitude spectrum vs. shotpoint

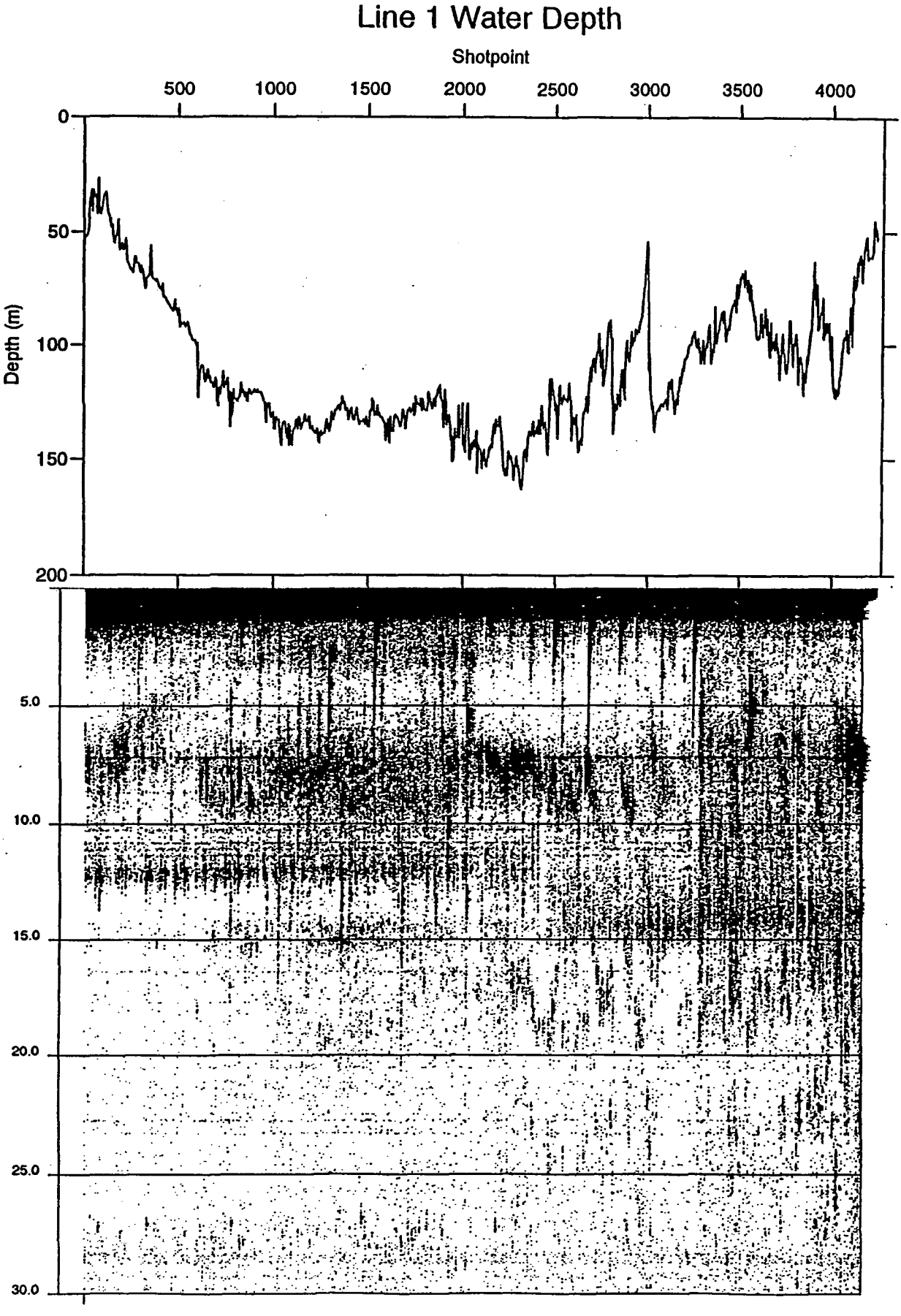


Fig. 5.4 c) Station 1A frequency amplitude spectrum vs. shotpoint

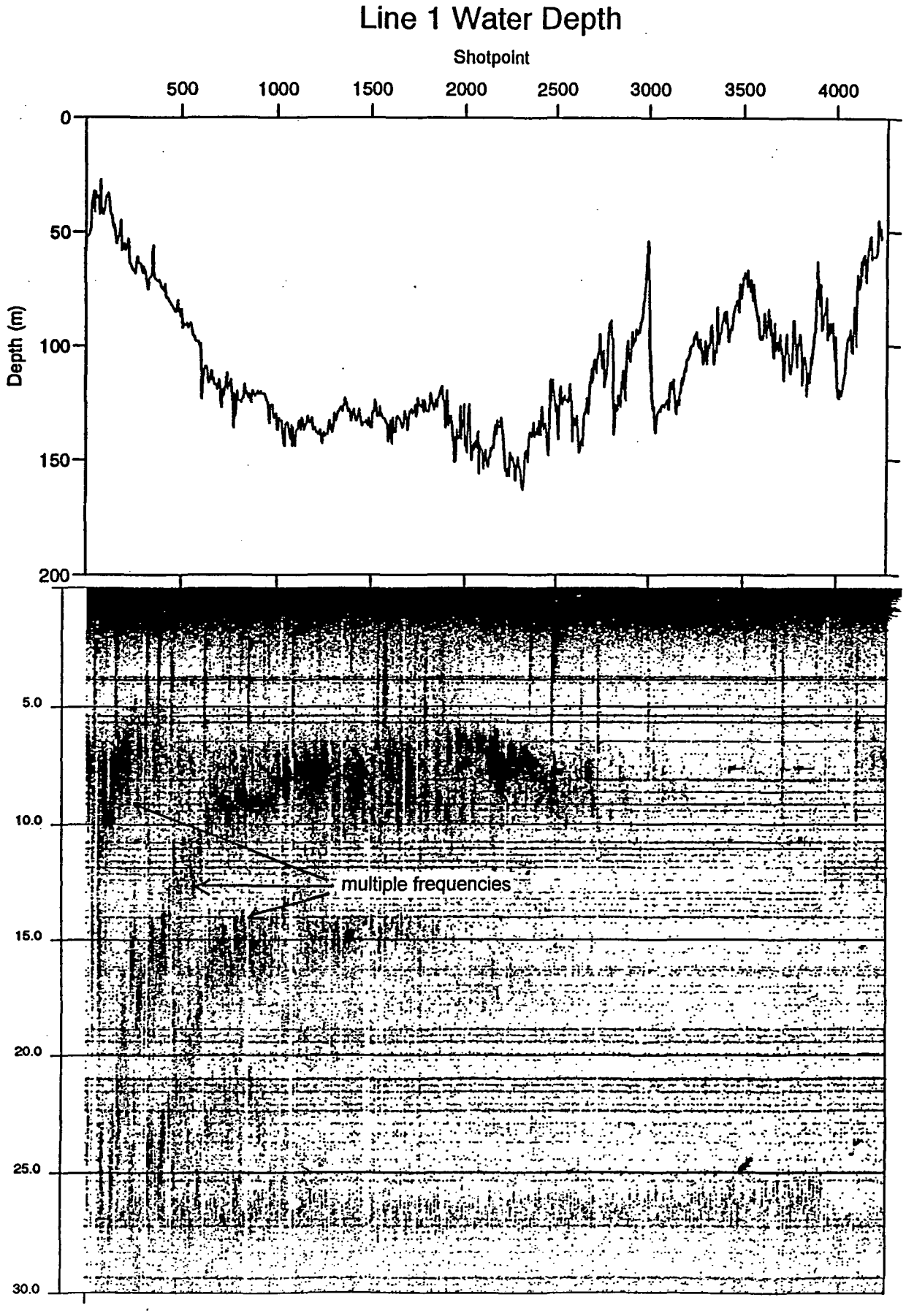


Fig. 5.4 d) Station 5A frequency amplitude spectrum vs. shotpoint

To determine the presence of water layer multiples, a correlation between frequency content and water depth was looked for. To do this, the frequency amplitude spectrum was taken for every tenth trace along the line using the program F-T (see Appendix 1). These were plotted as variable area traces against shotpoint. The water depth as determined from the survey ship for each shotpoint was extracted from the trace headers (program WATERD, Appendix 1) and also plotted on the same distance scale (Fig. 5.4).

Finally, a number of test panels were filtered using a rangeof band pass frequency windows in order to find the range of frequencies which would optimise the record section clarity. The program PANEL (see Appendix 1) was written by the author for this purpose. An example of this is shown in Fig. 5.5.

5.2.3 b) Frequency content of the data

In many of the individual amplitude spectra there were three large peaks (Fig. 5.3). The first and largest of these between 0 and 2.5 Hz could be ascribed to low frequency microseismic noise (e.g. from tree motion and waves on the shore), obviously incoherent from an unfiltered record section (Fig 5.6). This noise does not appear strongly on data from station F01A due to the low frequency response of the geophones used there (low frequency cutoff ~ 4.5 Hz)

The cause of the other two peaks, however, was not so obvious. It is clear that for some of the data, especially for stations F01A and 5A (Figs. 5.4b and 5.4d), there is a strong correlation between the variation of peak frequency and water depth with offset.

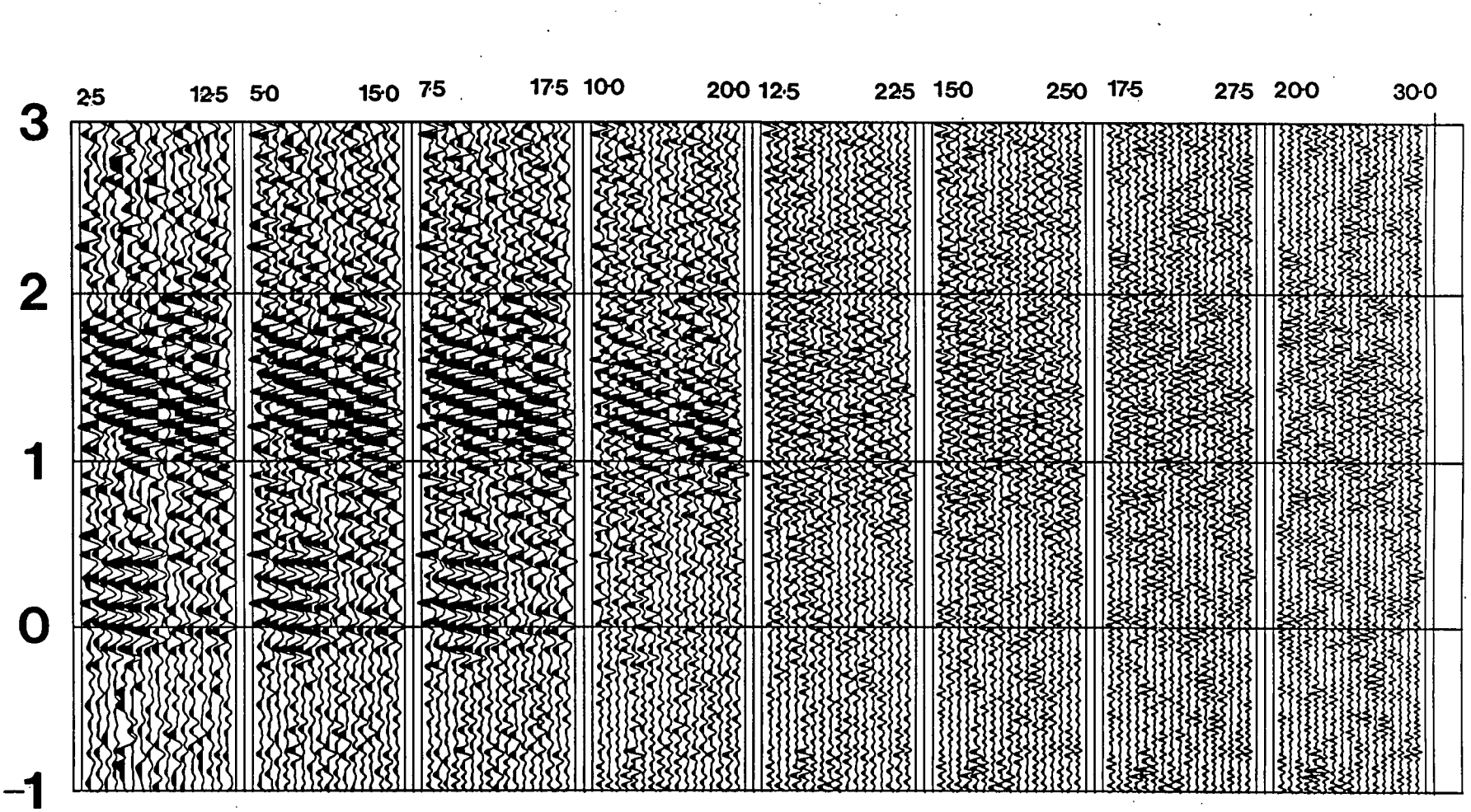


Fig. 5.5 A small panel of traces from station 5A, distance 120 km, band pass filtered with different windows

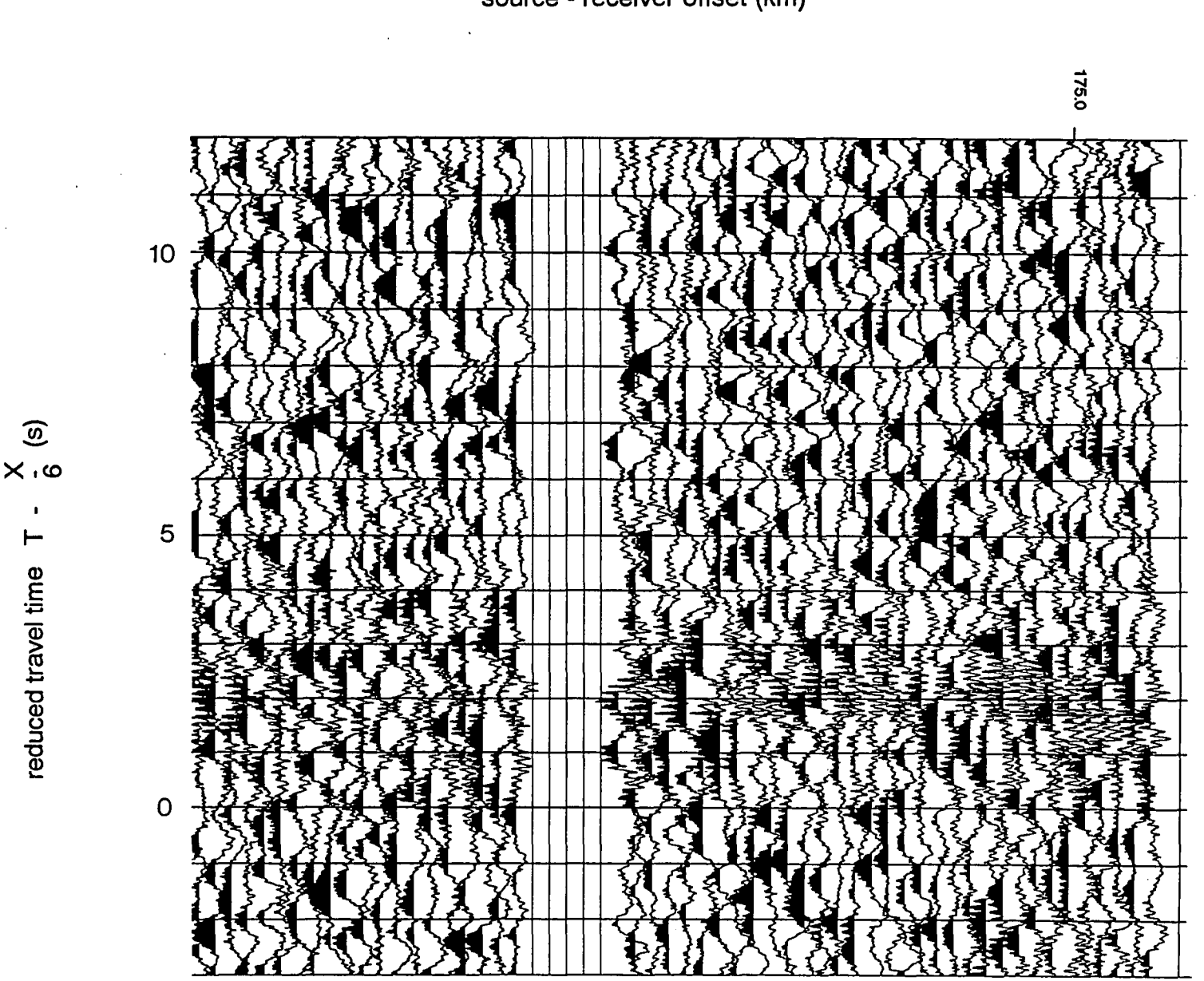


Fig. 5.6 Example of raw data from station 101P

Using the formula

$$n^{th} \text{ multiple frequency} = \frac{(2n - 1) \times V_w}{4h}$$ where h = water depth, V_w = water vel.

it can be shown that two of the varying peaks in Fig. 5.4b are the second and third water layer multiples (see overlay). The first can be seen at the far left-hand edge of the figure, the rest of it merging with the low frequency noise.

In addition to these variable frequency peaks, bands of relatively constant frequency can be seen (Figs 5.4 a,b and c). Two prominent ones can be seen on Fig. 5.4c (station 1A) at frequencies around 12 and 14 Hz. These may be caused by multiples in layers beneath the receiver of around 200 - 250 m thickness. Although no reflections from such layers are directly observed in the wide angle dataset, indications of sill-like features in the upper crust are seen in the normal incidence data (see Chapter 6, section 2).

The major frequency content of the data appears to lie in the range 5 to 20 Hz. This was confirmed by the band pass filtered data panels (Fig. 5.5), with no coherency being observed above around 20 Hz. Although the far field source signature (Fig. 4.6) has a relatively flat frequency spectrum up to around 75 Hz, the high frequency energy is cut out by the demodulating and anti-alias filters. Even so, little of the mid-range frequency energy between 20 Hz and 40 Hz reaches the land based receivers. Significant tuning seems to be occurring at low frequencies through water layer multiples.

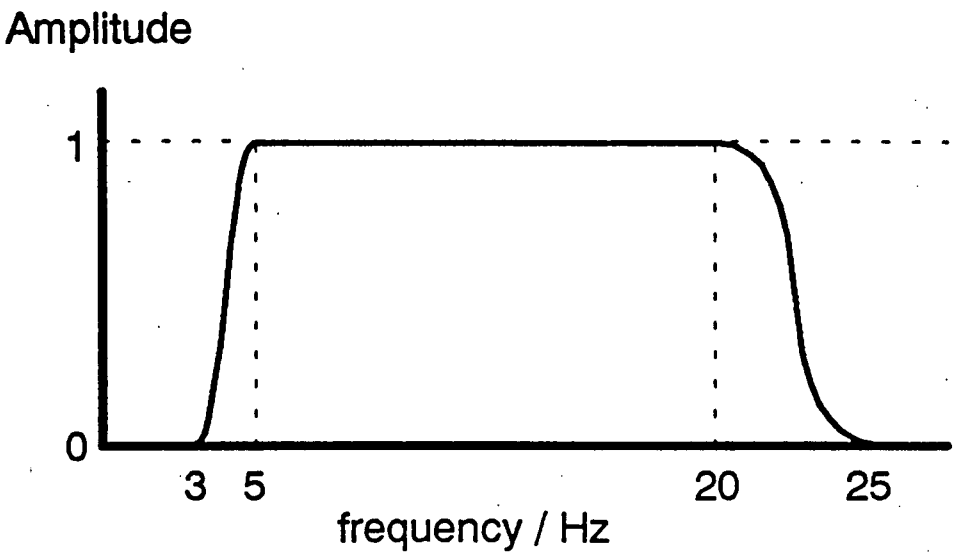


Fig. 5.7 Hanning band pass filter window used to filter all the data.

5.2.3 c) Band pass frequency filter

All the data were band pass frequency filtered using a Hanning window with parameters based on the conclusions reached in the frequency analysis, i.e. all frequencies outside the range 5 - 20 Hz were to be removed. The design of the filter is shown in Fig. 5.7. The actual 3dB points are 4 and 22 Hz, giving a little leeway at either end. The filter was applied with the program BPFILT written by the author (Appendix 1).

5.2.4 Field statics (Stage 3)

A field static correction was made to each trace reducing both the shotpoints and receiver location to a datum in an attempt to remove the effects of the water layer on primary (non water layer multiple) arrivals. The program used for this was STATICS (see Appendix 1), written by the author. The water depth measured by the survey ship for each shotpoint varied between 30 and 160 m with an average of 110 m. The source point datum was taken at a depth of 170 m beneath the surface. Since the source was towed at a depth of 15 m, this also had to be taken into account. The receiver datum was taken at sea level.

In order to calculate the static correction, i.e. the *delay time*, it was necessary to estimate the velocity of the media above and below the datum. A water velocity of 1.45 km s^{-1}, a sea bed rock velocity of 4.8 km s^{-1} and a near surface land bedrock velocity of 5.8 km s^{-1} were assumed. The delay time calculation is dependent on the ray parameter p and is therefore different for arrivals of different apparent velocity. To apply a single correction value to each data trace, a certain value for the ray

parameter had to be assumed. The ray parameter chosen was based on an arrival with apparent velocity 6.5 km $s^{-1}(=1/p)$. This is approximately the mean apparent velocity of crustal diving rays estimated from the data sections. The formula used for calculation of the source static correction is as follows:

$$Ts_{source} = p \times \frac{(\text{water depth} - \text{source depth}) \times \sqrt{\frac{1}{p^2} - V_{water}^2}}{V_{water}} + p \times \frac{(\text{receiver elevation} - \text{receiver datum}) \times \sqrt{\frac{1}{p^2} - V_{seabedrock}^2}}{V_{seabedrock}}$$

That used at the receiver is :

$$Ts_{receiver} = p \times \frac{(\text{receiver elevation} - \text{receiver datum}) \times \sqrt{\frac{1}{p^2} - V_{bedrock}^2}}{V_{bedrock}}$$

The maximum source static correction applied was 0.1 seconds while the largest receiver correction used in this study (constant for each station) was 0.02 seconds for station 1A.

5.2.5 Processing Branch A

At this point, the processing sequence forked into two separate branches (Fig. 5.1), the first leading towards velocity filtering in f-k space.

5.2.5 a) Binning data (Stage 4A)

Line 1 was shot in three parts with a small overlap at each end so that at these overlaps, the data density was doubled. For the off-line stations, trace spacing at the shortest offsets also became very small. In order that processing requiring a constant trace spacing such as f-k analysis and velocity filtering could be carried out, the data were binned. Binning was chosen over spatial resampling as the latter is far heavier on computer resources and gives less reliable results with a wide angle dataset. The program used for this was TRACEBIN, written by the author (see Appendix 1).

Within each line segment, the trace spacing was relatively constant for the in-line stations. A bin width of 75 m was therefore used to preserve density of data. For most of the line, this meant that only one trace fell into each bin interval. The distance error introduced by this was too small (max. ±37.5 m) to produce scatter (i.e. timing errors · 1 sample) in arrivals with apparent velocities in the range 4.3 to 10.0 km s^{-1} for data with a sample interval of 5 ms, and 3.3 to 30.0 km s^{-1} for a sample interval of 10 ms. All the arrivals of interest on the section lie within the narrower of these ranges.

5.2.5 b) Resampling in time domain (Stage 5A)

Having band pass filtered and binned the data, resampling was carried out in order to regularise the sample rate for the Geostore data (see section 5.2.2), and reduce it to the minimum necessary to preserve the frequency content without aliasing. Since, after band pass filtering, the

data contained no signal with frequency greater than 25 Hz, the new sample rate was set at 50 samples per second (sample interval = 20 ms). This had the effect of decreasing the volume of data by one half for the PDAS data and three quarters for the Geostore data, making it easier to handle and also allowing much faster plotting. The program, written by the author, to perform the resampling was called EXTR (see Appendix 1).

5.2.5 c) F-K spectra and velocity filtering (Stage 6A)

A 2-D Fast Fourier Transform was applied to a 512 x 512 square array of data in order to examine its f-k amplitude spectrum, using the program F-K, written by the author (see Appendix 1). At a sample rate of 50 samples / second and trace separation of 75 m, the window dimensions were 10.24 seconds x 38.4 km. Since the data was already reduced at 6 km s^{-1}, all the P-wave arrivals within the distance range could be included in the time window. At near offsets, S-wave arrivals also appear on the spectra. An example is shown later, in Fig. 6.5a.

The effect of reduction on the f-k plot is to cause arrivals with apparent velocity equal to the reduction velocity (6 km s^{-1} in this case) to lie along the vertical axis. Fig. 5.8 shows some apparent velocities as they would appear on the reduced f-k plot.

In an attempt to remove the high amplitude S-wave arrivals which at short offsets interfered with sub-critical reflection phases, a pie-slice filter was applied to a window of data from station 101P in the f-k domain. A pie-slice filter with cosine taper was chosen as it would cut out all data outside a given velocity range and was easy to define and apply. The filter was applied to the real and imaginary parts of the 2-D Fourier transform in order

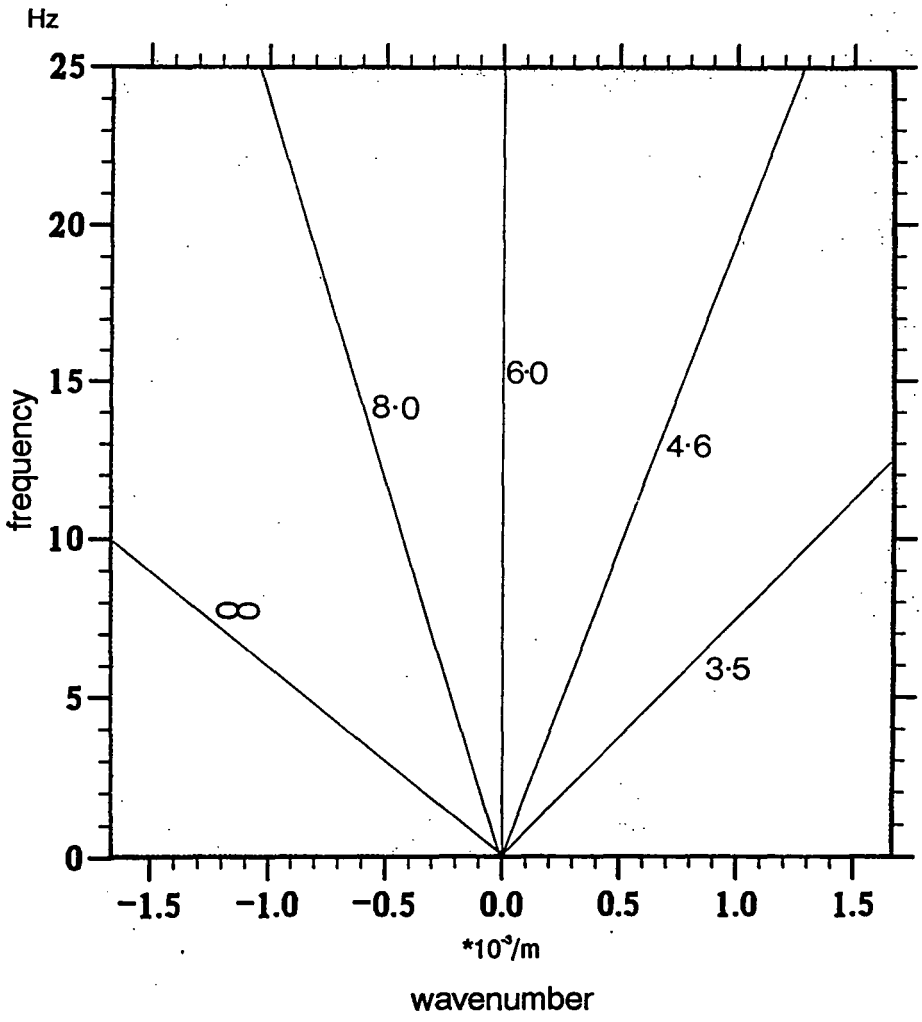
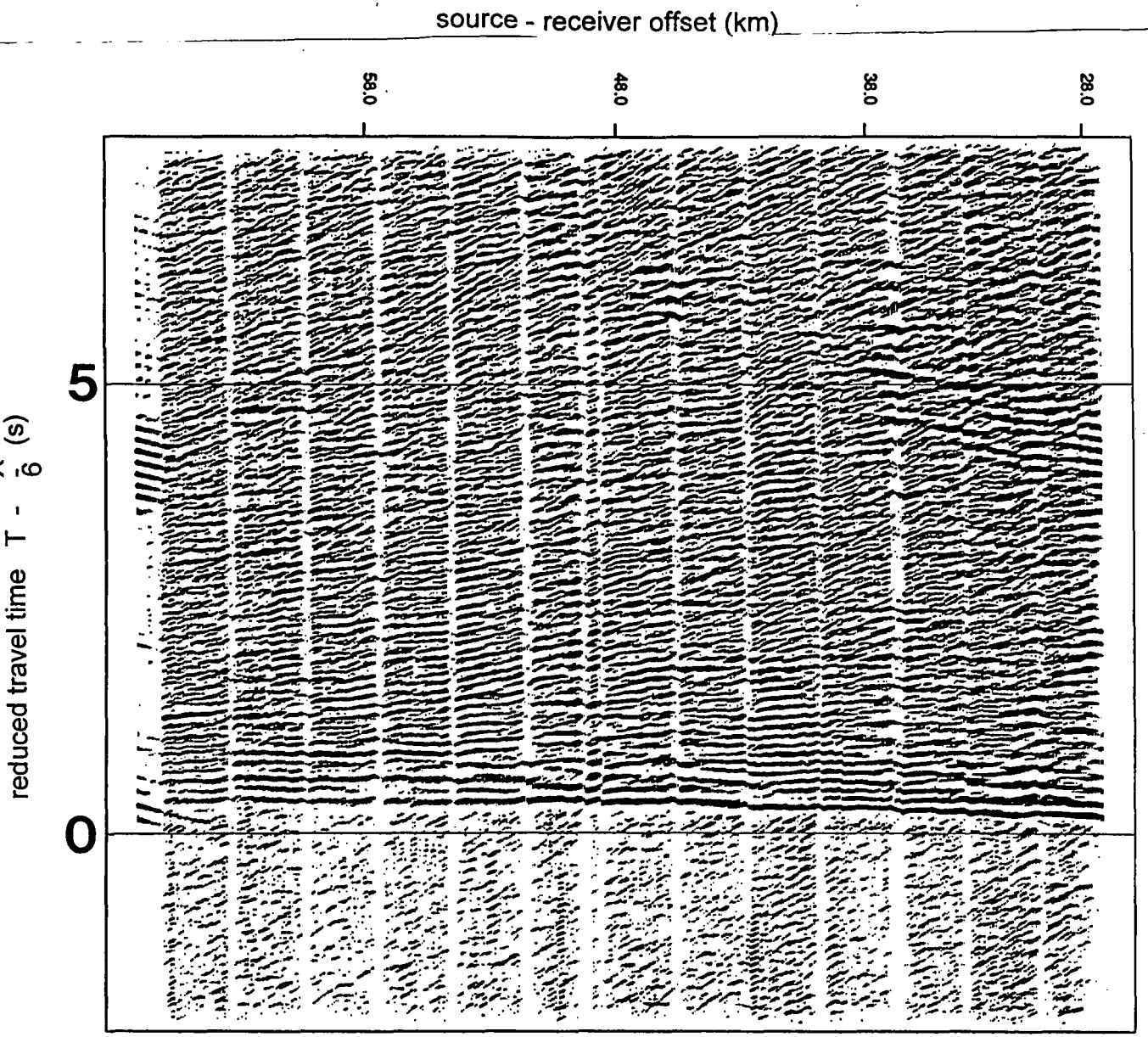


Fig. 5.8 Some examples of apparent velocities on an f-k spectrum of data reduced at 6 km s^{-1}



b) after velocity filtering

to filter out all arrivals with apparent velocities below 4.5 km s^{-1} whilst retaining virtually all the higher velocity arrivals. The results of this are shown in Chapter 6 (Figs. 6.4 and 6.5).

An attempt was also made to filter out the S-wave arrivals from the previous shot which interfered with the P-wave arrivals between offsets 170 and 320 km. Four adjacent windows were required to cover this distance range. A pie-slice filter was again applied to each window, chosen to filter out all arrivals with apparent velocity below 5.0 km s^{-1}. The f-k spectra and results of velocity filtering for station F01A are shown in Chapter 6 (Figs. 6.9, 6.10 and 6.11). The success of this filtering was limited as much of the noise was smeared out at the cutoff velocity rather than being removed. This could be due to the steepness of the taper and also to the fact that, due to small timing differences between shots, the arrivals from the previous shot were in short, broken segments rather than continuous (Fig 5.9). Such broken segments introduce step functions in the x domain of the data which produce side lobes or 'ringing' in the wavenumber domain. If these side lobes are then filtered out, the result is ringing in the x domain. A filter in f-k space in which the cut off taper is too steep has a similar effect of producing side lobes in the data in x-t space after the inverse Fourier transform.

5.2.6 Processing Branch B: Stacking with linear moveout (Stage 4B)

As an alternative to binning and f-k filtering, a five-trace rolling stack with *linear moveout* was applied to the data in an attempt to enhance arrivals of a given velocity. The program used for this was STACK, written by the author (see Appendix 1). In stacking with linear moveout, a

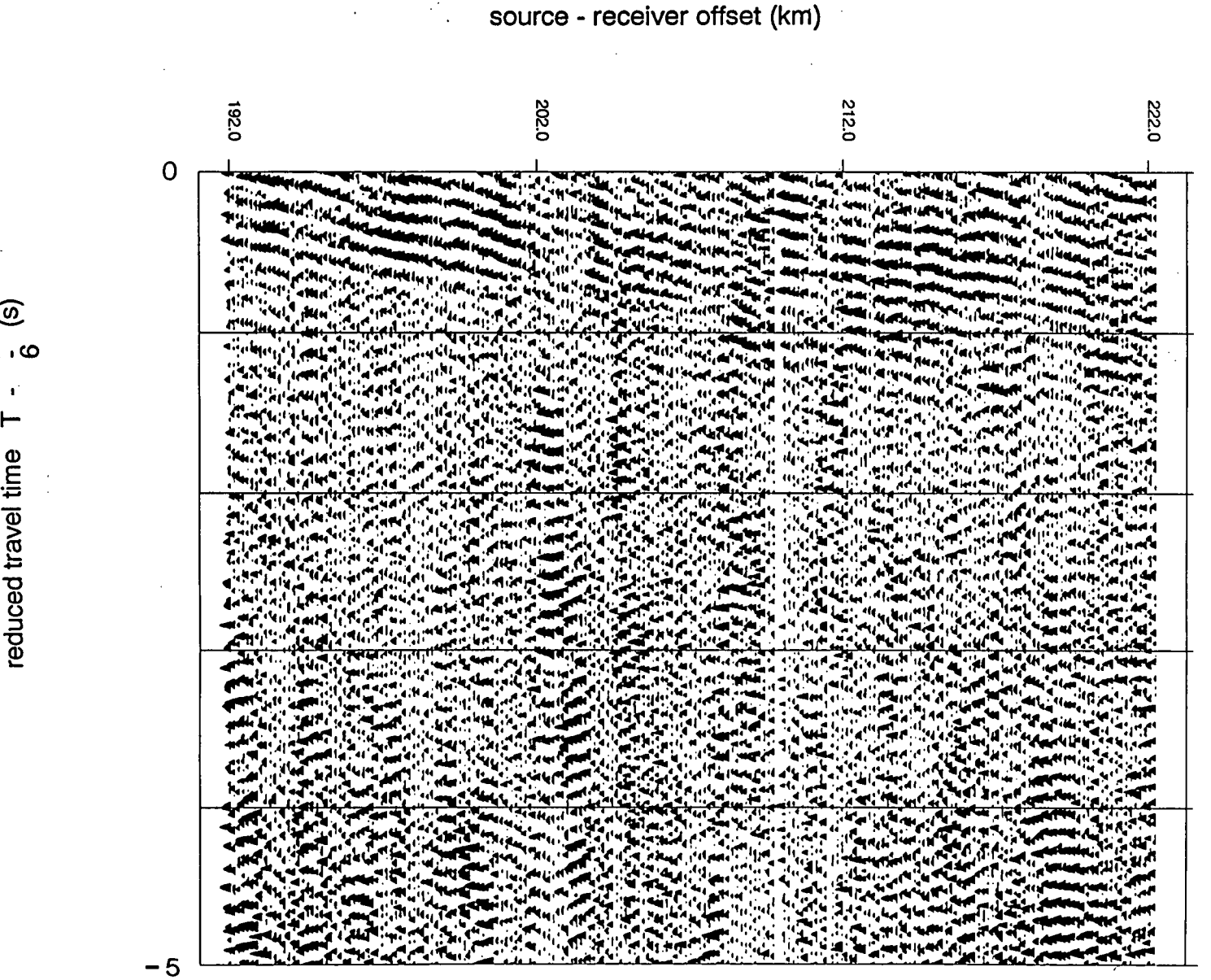


Fig. 5.9 Example of S-wave arrivals from the previous shot (Station F01A)

correction is applied to each trace within the window before stacking. The correction varies linearly with offset from the window origin. This is equivalent to re-reducing the data within the window at the selected velocity before stacking (see section 5.2.2). The velocity chosen was 6.5 km s^{-1}, approximately the mean crustal velocity as estimated from the record sections. Due to the short width of the stack window, the range of velocities stacked coherently was large enough to cover all the crustal diving rays. Smaller windows were found to be ineffectual while larger ones tended to distort the data beyond usefulness (Fig. 5.10). After stacking, this data was also resampled in time as for the binned data (section 5.2.5b).

5.2.7 Plotting

The data was plotted with a program initially written by C. Prescott at Durham University, modified and adapted by C. Peirce (DAZZLE, Appendix 4), which uses the UNIRAS graphics library. In general, true relative amplitude sections were plotted, the whole dataset being scaled according to an estimate of the mean amplitude on the section. However, in the data from the southern end of Line 1, the amplitude dropped off severely at large offsets so trace-normalised sections were plotted in which all the traces were scaled individually so that the maximum amplitude in each trace was constant across the section. In addition, sections with an a.g.c. applied were plotted in order to bring out weaker arrivals. However, since an a.g.c window of 100 ms was used, weak arrivals closer than this to strong ones were suppressed.

For interpretation purposes, the data were plotted as wiggle traces with positive variable area fill. Twenty seconds of data were plotted on a 24

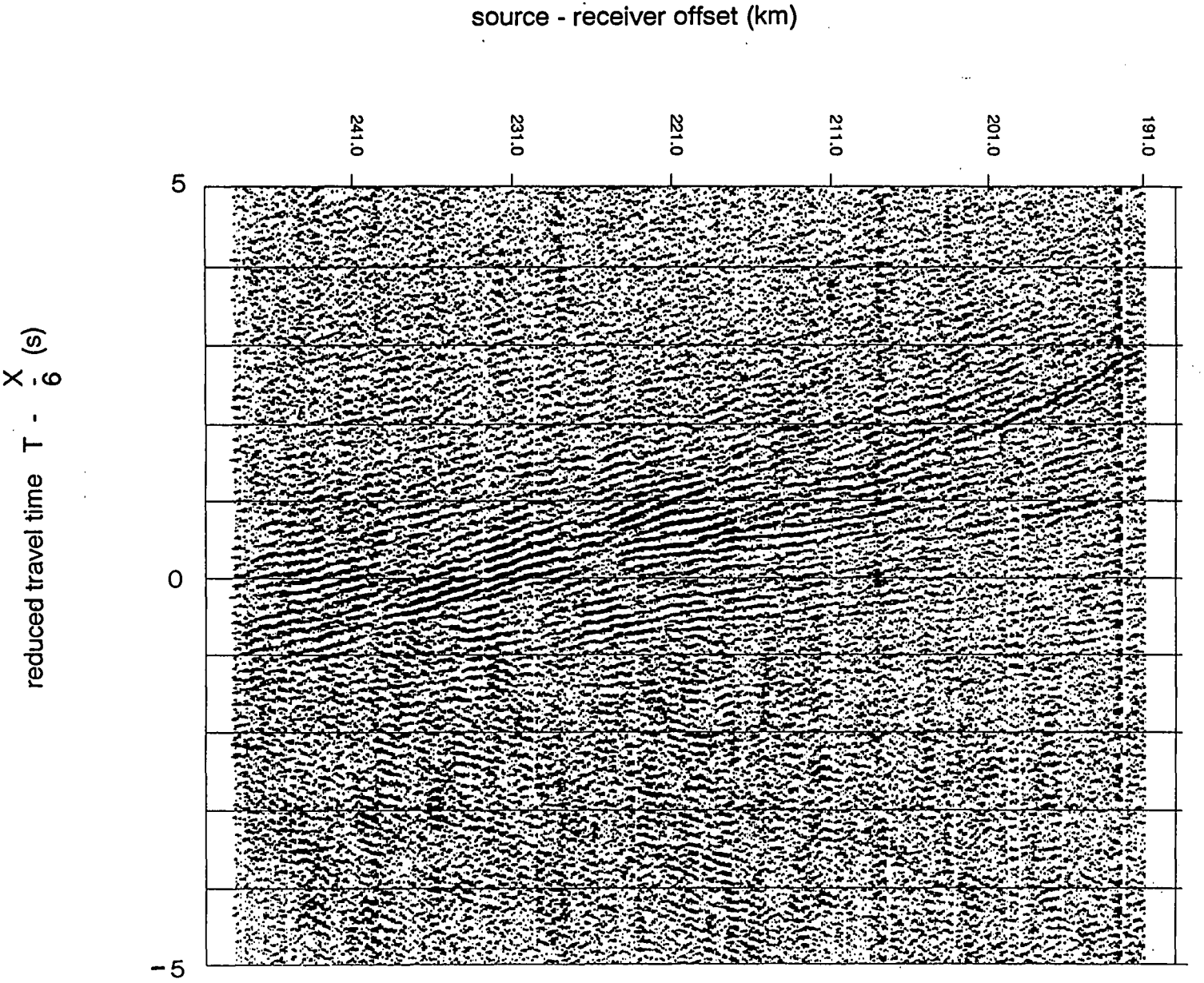


Fig 5.10 Sample of the record section from station 1A showing the effect of stacking with different windows:
a) no stacking

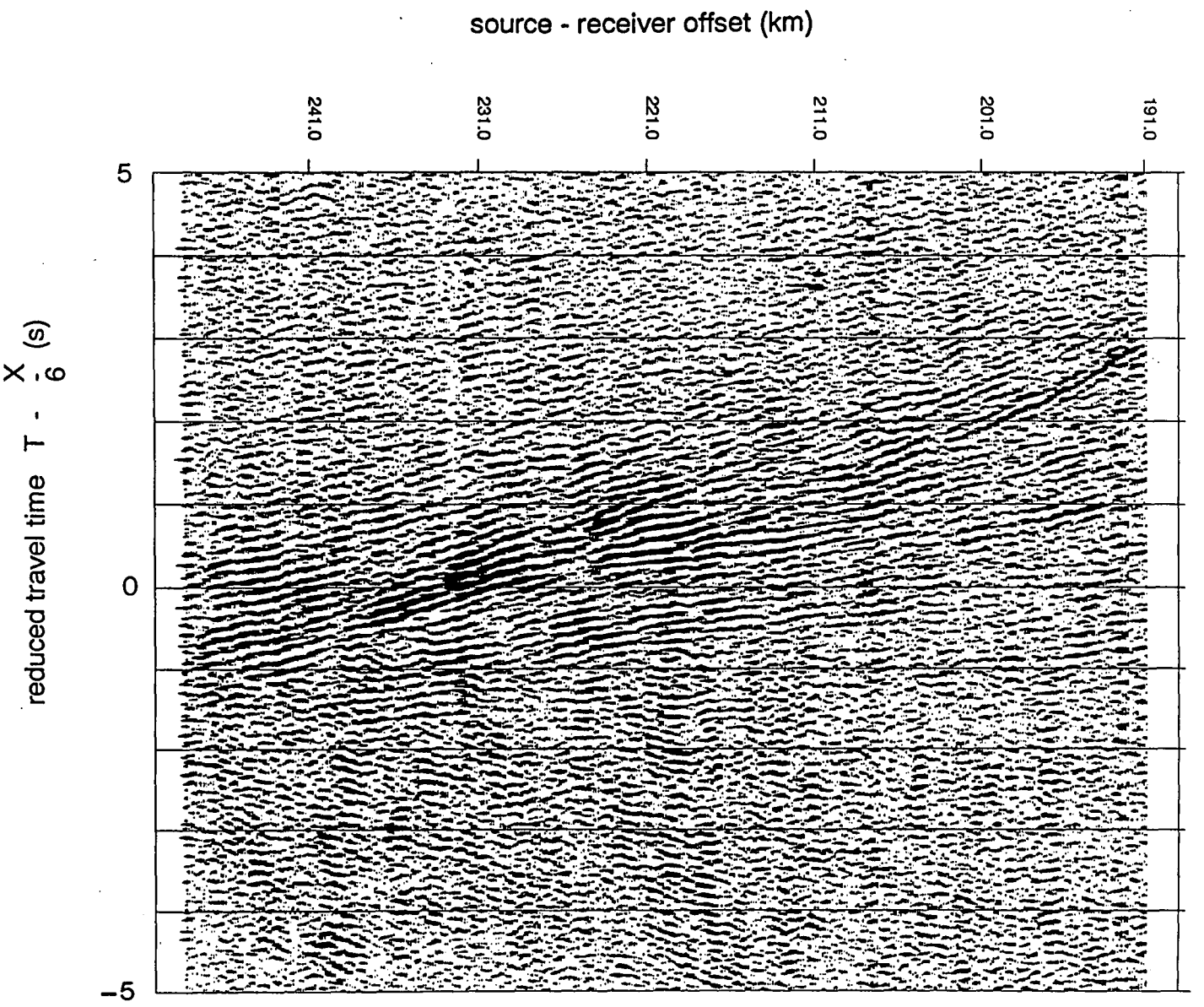


Fig 5.10 b) 3 trace rolling stack with 6.5 km s^{-1} linear moveout

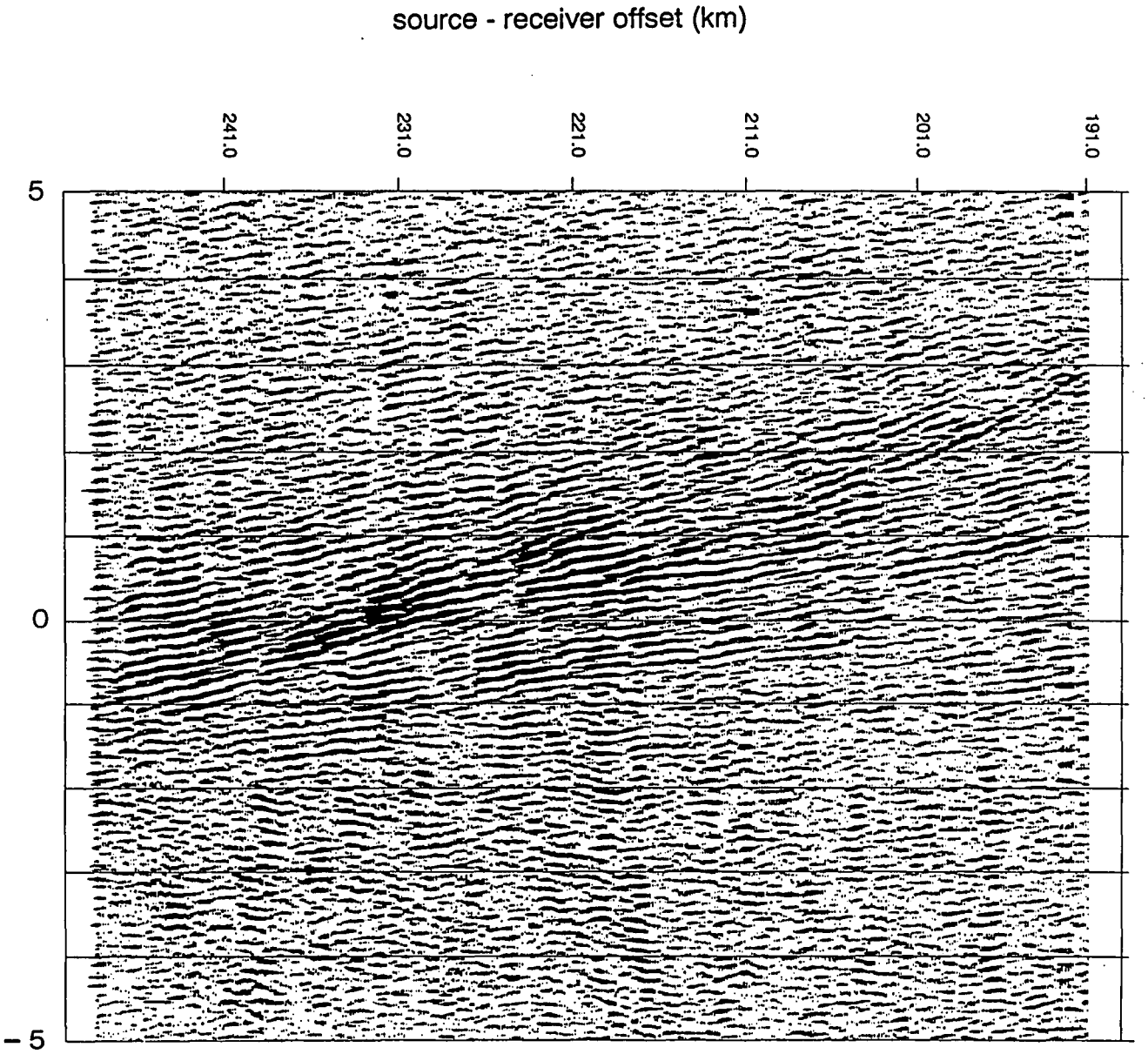


Fig 5.10 c) 5 trace rolling stack with 6.5 km s^{-1} linear moveout

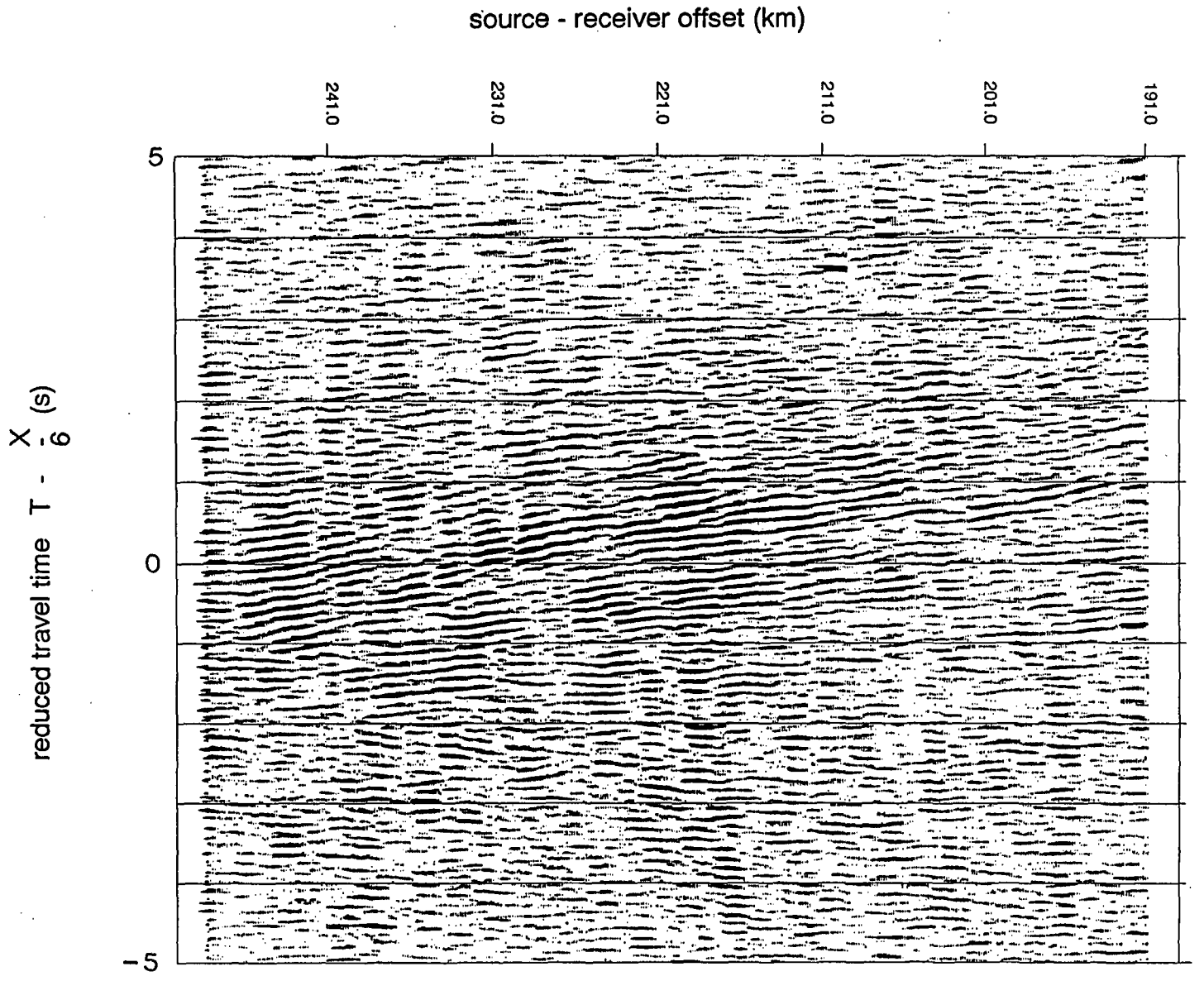


Fig 5.10 d) 10 trace rolling stack with 6.5 km s^{-1} linear moveout

inch electrostatic Versatec plotter with a distance scale of 150 m/mm. For smaller plots, the wiggle traces became so densely spaced that they obscured the data so only the variable area was used. For A4 sized plots, every 5th trace was plotted for similar reasons, with a distance scale of 1600 m/mm. The software allowed the overall amplitude scaling, threshold level and high cutoff to be controlled so that low amplitude noise could be cut out. These parameters were adjusted on a trial and error basis to give the clearest plot.

5.3 Normal Incidence Data

All the normal incidence processing was done by the contractors PRAKLA SEISMOS and BIRPS. The processing sequence shown in Table 5.3 is taken from the information panel of the final stacked paper sections of the normal incidence data sent to Durham by BIRPS. These sections were too large to handle easily, with a distance scale of 50 m/mm so a much smaller plot with an a.g.c. applied, also acquired from BIRPS, was used to get a better impression of the overall pattern of reflectors. The digital data was also acquired from BIRPS and plotted at a similar scale with true relative amplitude.

Table 5.3 Normal Incidence processing sequence: Final Stack

1. INPUT: TAPE FORMAT SEG-D, 9-TRACK, 6250 BPI
   SAMPLE RATE 4 MS, 60 TRACES
2. S.O.D. CORRECTION -41 MS
3. FREQUENCY FILTER 5-48 HZ, SLOPE 24 DB/OCT
4. EDITING
5. RESAMPLING 4 MS TO 8 MS
6. AMPLITUDE CORRECTION FOR SPHERICAL DIVERGENCE
7. WHOLE TRACE EQUALIZATION, GATE: 3000 - 17000 MS
8. RECEIVER ARRAY SIMULATION BY MIXING IN COMM.-SOURCE DOMAIN
   TIME VARIANT NO MIX:TO 0.5 S; FULL MIX:4 S - END (1.2.4.2.1)
9. DECONVOLUTION:

DESIGN:	FIRST GATE /	SECOND GATE
ON SHOTNEAR TRACE :	500 - 6000 MS /	6000 - 14000 MS
ON SHOTFAR TRACE :	3000 - 6000 MS /	6000 - 14000 MS
APPLICATION:	0 - 5000 MS /	7000 - END
OPERATOR LENGTH:	392 MS /	528 MS
PREDICTION INTERVALS:	32 MS /	48 MS
PREWHITENING FACTOR	1%	

10. NMO CORRECTIONS
11. MUTING
12 HORIZONTAL STACKING 2400 %
13. SHOT AND STREAMER CORRECTION 15 MS
14. FK FILTER: TYP: PASS; POLYGONAL
    (2.5 - 50 HZ, +/- 60% WAVENUMBER, +/- 18 MS/TRACE)
15. DECONVOLUTION:

	FIRST GATE /	SECOND GATE
DESIGN:	500 - 6000 MS /	6000 - 14000 MS
APPLICATION:	0 - 5000 MS /	7000 - END
OPERATOR LENGTH:	392 MS /	528 MS
PREDICTION INTERVALS:	32 MS /	48 MS
PREWHITENING FACTOR	1%	

16. TV FREQUENCY FILTER: 10/24 -40 /24 HZ/ DB/OCT AT 3S
    5/24 -20 /24 HZ/ DB/OCT AT 20S

6. Data

6.1 Wide Angle Data

6.1.1 Introduction

As stated in chapter 5, all the data shown here are plotted as variable area with no wiggle. A distance scale of 1600m/mm (1:1600000) was chosen so that each data section would fit on one A4 sheet. Much larger plots (1:400000 and 1:150000) were used for interpretation and four of these have been included in Appendix 5. Distances are measured from the receiver and the sections are plotted from south to north, left to right. All the data shown are the vertical component.

True relative amplitude sections were produced for all the datasets considered. In some cases the amplitude fell off so markedly with distance that little could be seen at the greater offsets without completely saturating the near offset traces. For these datasets normalised sections were also produced in which the peak amplitude in each trace was normalised to a constant value.

In picking arrivals on the section, as little initial interpretation of individual arrivals was done as possible so that preconceptions would not be carried over into the modelling process. An attempt was made to ignore multiples of both long- and short-period. These were identified as arrivals of similar shape and character to phases appearing earlier in the section. The short period multiples appeared to have frequencies similar to those expected for mode 2 water layer multiples (see Chapter 5, section 2.3b),

i.e. around 7 Hz. The longer period multiples, observed mainly at shorter offsets, had periods of a few seconds. Some of these are pointed out in the data sections in Appendix 5.

Emphasis was put on picking first arrivals and high amplitude later arrivals although a number of low amplitude phases were also identified by virtue of their coherency over short distances alone. No attempt was made to join together isolated arrivals with a continuous curve if intervening phases could not be seen.

6.1.2 Station 101P

Comparing the binned and stacked sections for this station (Figs 6.1 and 6.2), the stacking process has reduced the prominence of the white vertical stripes (caused by loss of data whilst off loading from the PDAS to lap-top computer in the field - see chapter 4) by smearing data across them. This smearing effect makes individual P-wave arrivals, especially the first arrivals, easier to see and pick but also has a tendency to delocalise and obscure the major packets of energy. Although this could be minimised by using a smaller stacking window, smaller windows were found to have little beneficial effect either. The window chosen was a compromise between enhancing the P-wave arrivals and delocalising the energy.

The near offset first arrival obviously has an apparent velocity of just below 6 km s^{-1} (a). This increases to 6 km s^{-1} at 50 km and remains roughly constant to 80 km where it peters out and is overtaken by an upwardly concave arrival (b). This can be traced back to the nearest offset traces where its amplitude becomes much lower. A much stronger arrival lies directly behind it (c), also upwardly concave but with an odd kink at

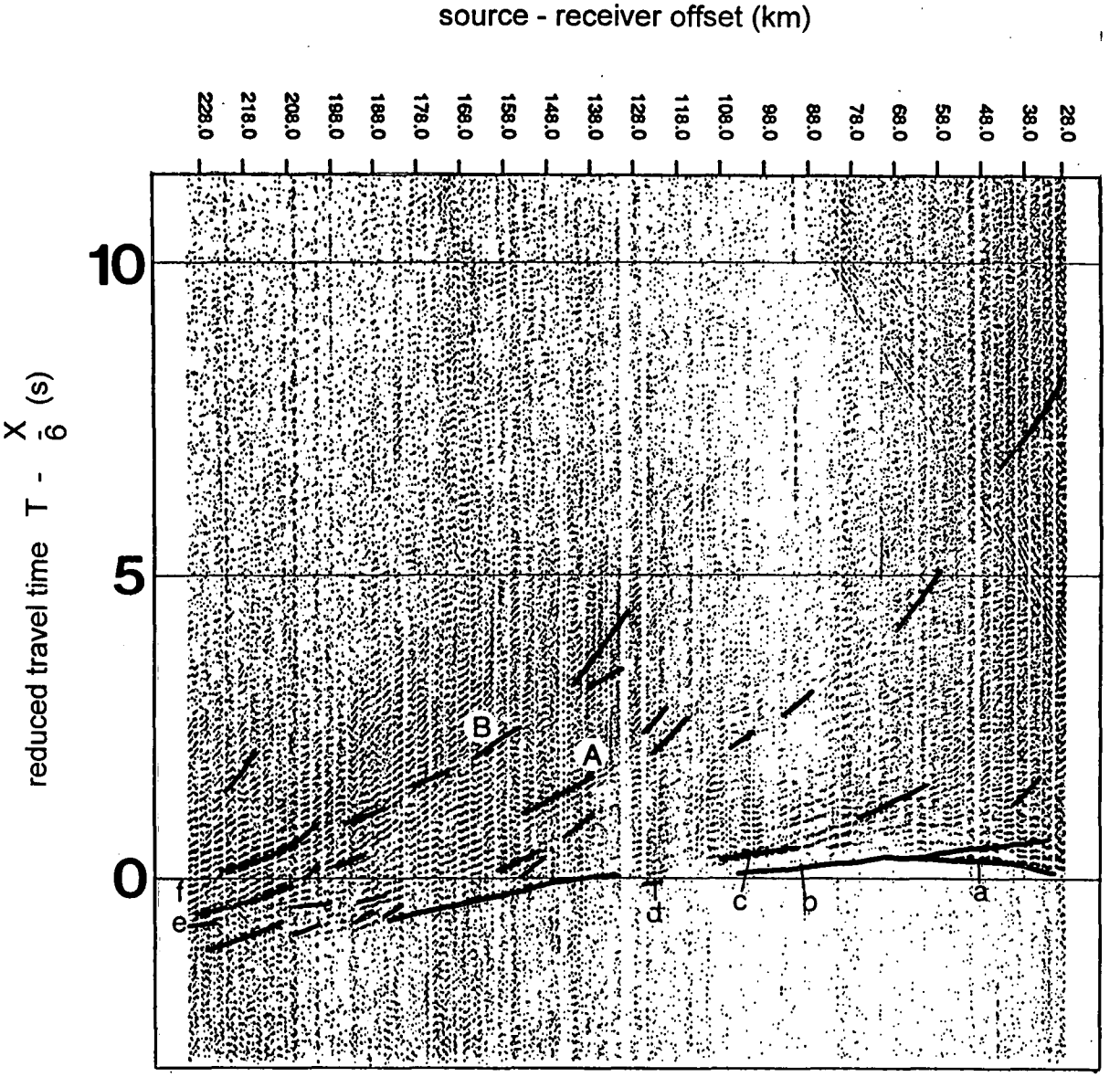


Fig. 6.1 BABEL Line 1, Station 101P: binned dataset, true relative amplitude

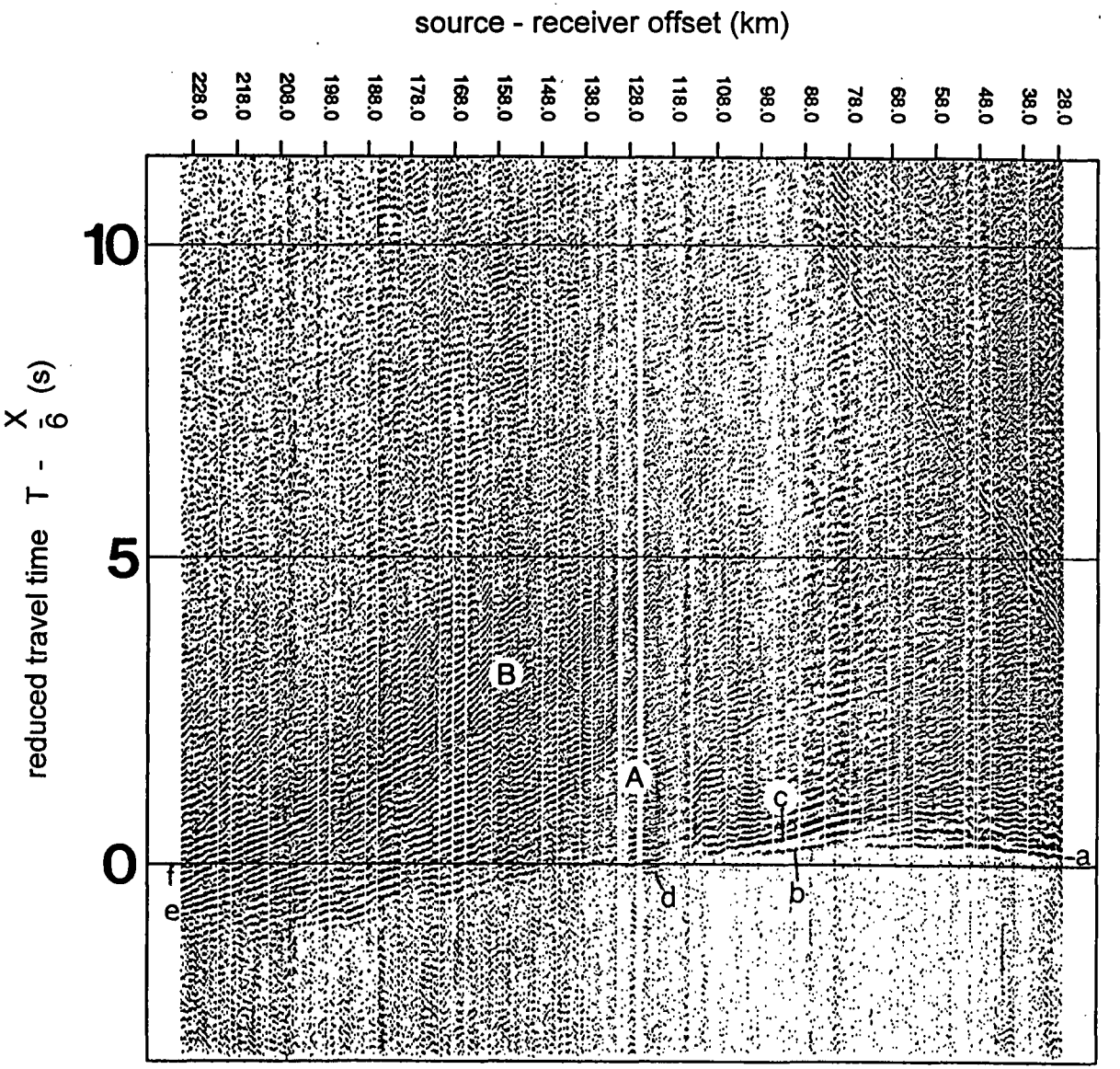


Fig. 6.2 BABEL Line 1, Station 101P: stacked dataset, true relative amplitude

about 85 km. This arrival appears to have higher curvature than earlier ones at near offset and takes over as first arrival at about 110 km. At about this offset all the arrivals steepen abruptly for a short distance. The corresponding shotpoints are directly above the Strömmingsbåden scarp system (Fig. 2.9). The arrivals do not have the characteristic hyperbolic shape of a diffraction and the effect seems to occur over the whole duration of the section. It is therefore thought to be a near surface effect associated with the scarp system, probably an incorrect removal of statics. Beyond this, between 120 and 130 km, an anomalously early first arrival is seen (d). At longer offsets, the first arrival increases in apparent velocity while generally decreasing in amplitude, fading out at 220 km.

Most of the energy at offsets beyond 120 km lies in two packets of arrivals, marked A and B in Fig. 6.1. The first (A) is prominent between 120 and 180 km. The general trend of the energy has higher curvature than one would expect of a reflection from a plane horizontal boundary. However, individual arrivals within it do not all run parallel to the trend of the energy packet or to each other, some having lower apparent velocities, others higher, but significant amplitude only within the region of the energy packet described.

The same is true to an even greater extent of the later energy packet (B) between 120 km and the farther end of the section. A number of individual branches can be made out and at the longest offsets, two major arrivals can be seen running parallel to each other about 0.5 s apart (e), (f). The earlier of these is not seen closer than 180 km from the station. A lower curvature, lower amplitude arrival can be traced out nearer than the upward turn of the major energy packet. This type of forking is also seen elsewhere within both energy packets.

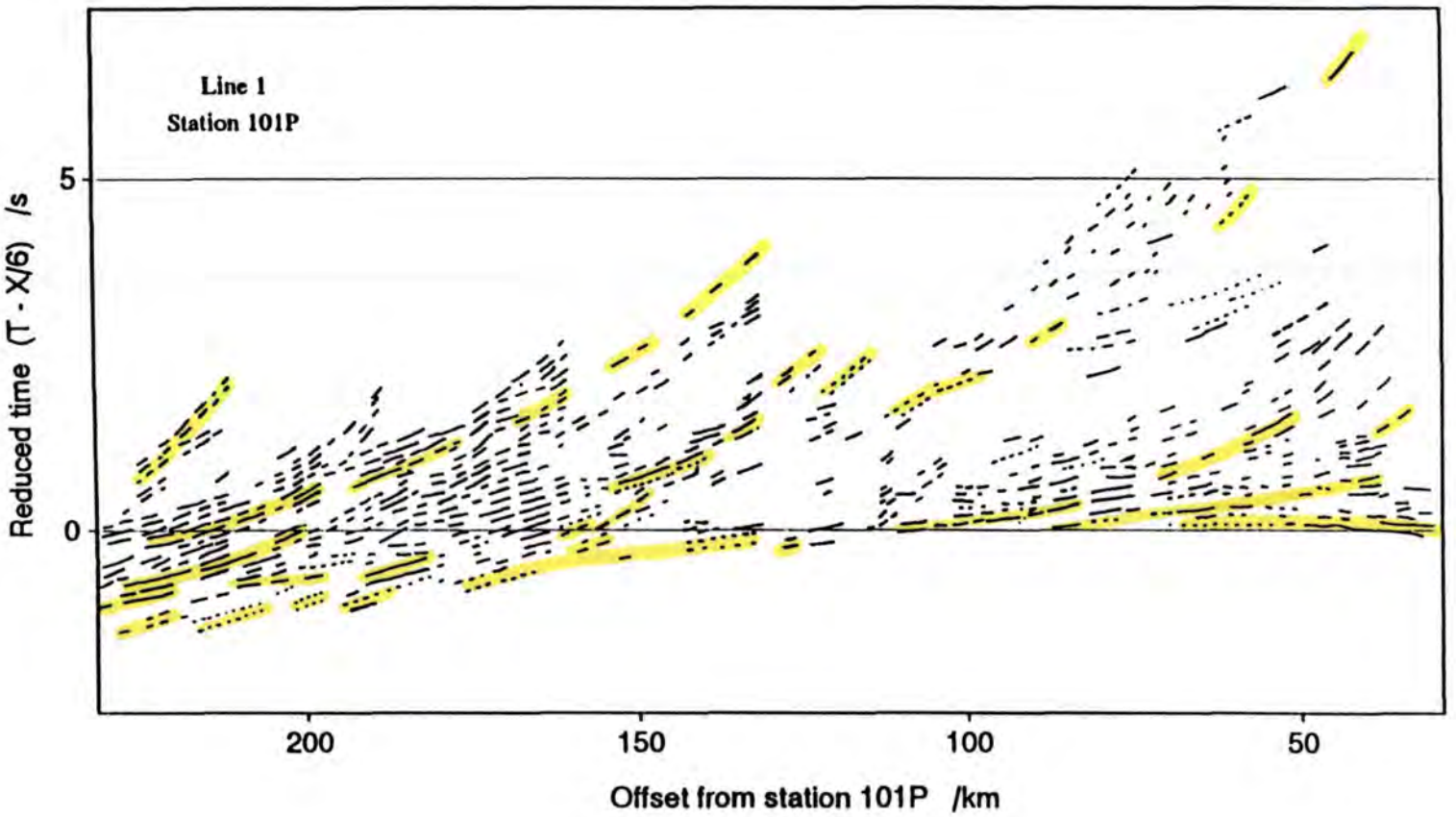


Fig. 6.3 BABEL Line 1, Station 101P: line drawing

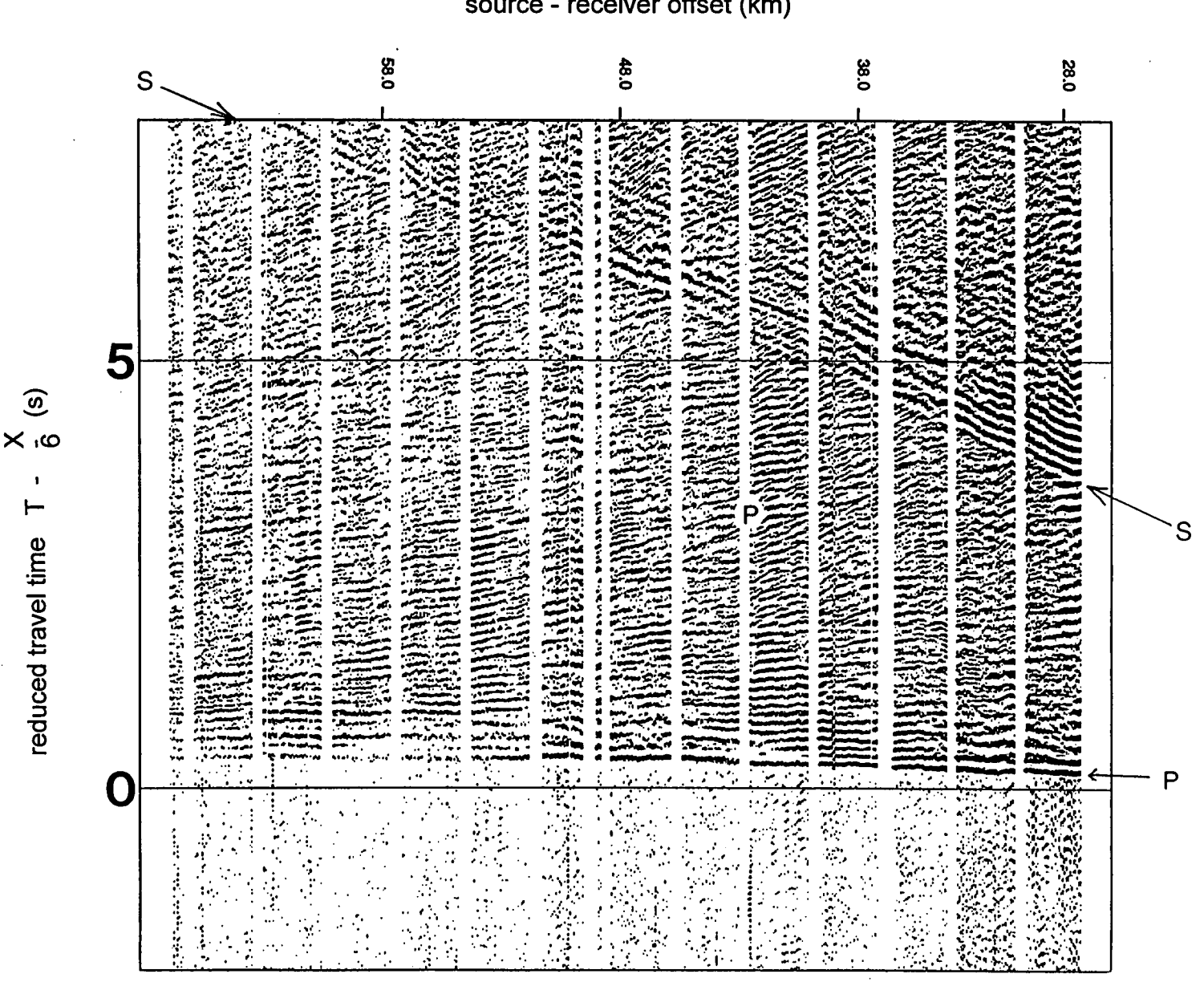


Fig. 6.4 BABEL Line 1, Station 101P: near offset data

a) before velocity filtering

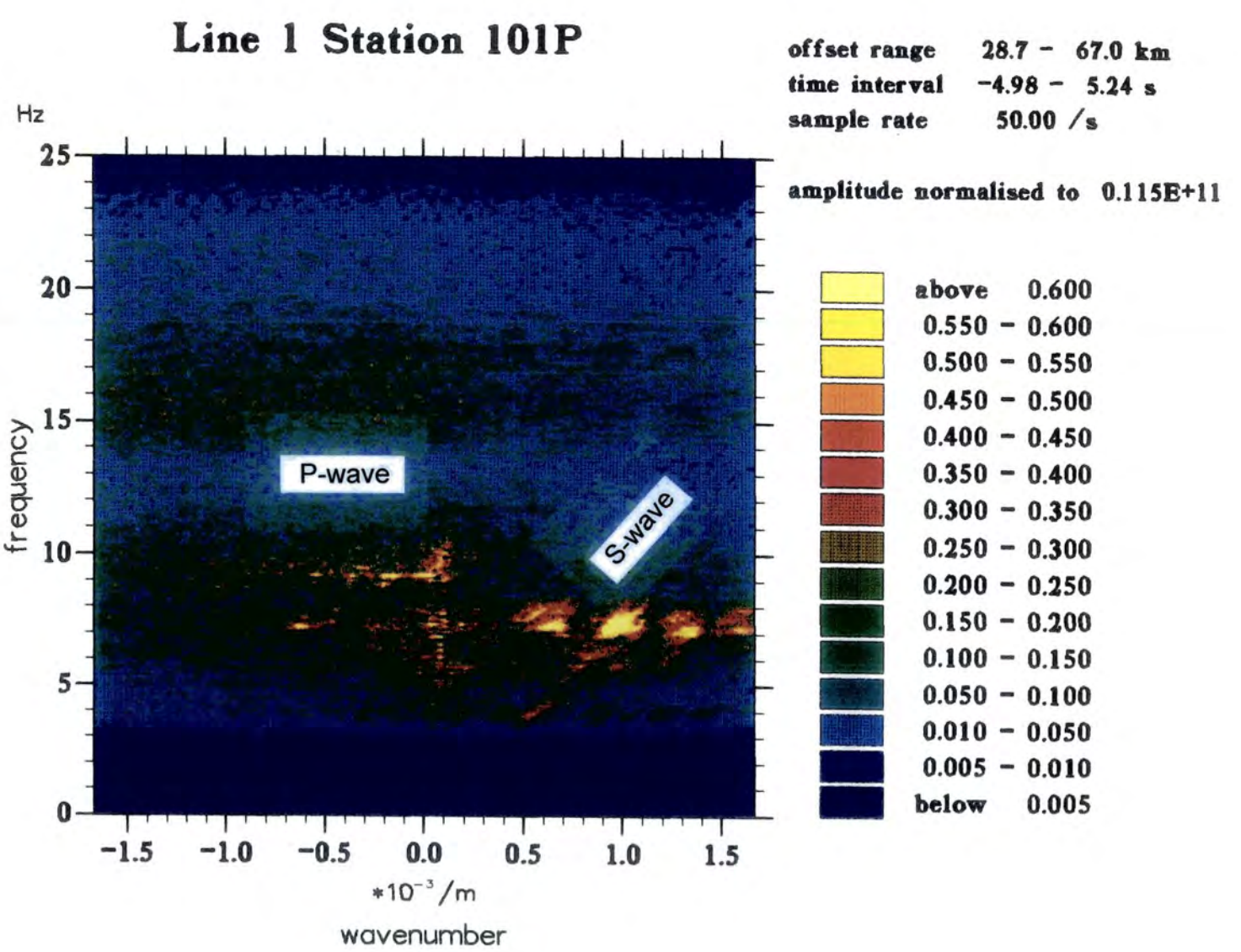


Fig. 6.5 a) BABEL Line 1, Station 101P: f-k spectra for the data shown in Fig. 6.4 before velocity filter

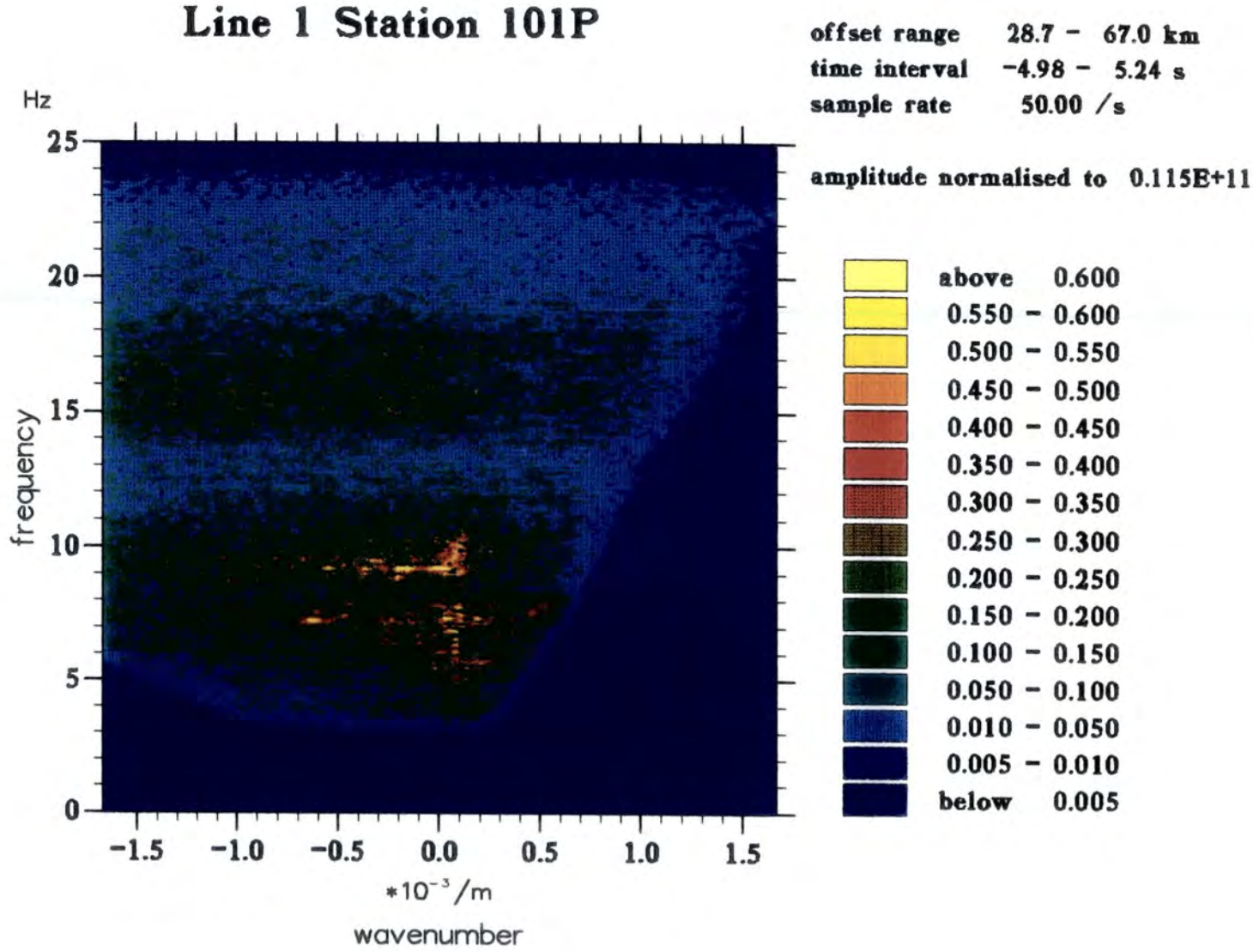


Fig. 6.5 b) BABEL Line 1, Station 101P: f-k spectra for the data shown in Fig. 6.4 with velocity filter applied

The overlay for Fig 6.1 shows the picks used for modelling while Fig. 6.3 shows a line drawing of the section with the picks highlighted.

There is some evidence for pre-critical reflections in the later part of the section at near offset, which is partly obscured by the strong S-wave arrival. An f-k spectrum of this region was examined and a pie-slice velocity filter applied to remove all arrivals with velocity below 4.5 km s^{-1}, as described in the previous chapter. Figs 6.4 and 6.5 show the original and the filtered data and f-k spectra respectively. Although most of the S-wave diving ray energy has been removed, the pre-critical reflected S-wave energy remains. These arrivals have apparent velocity very close to that of the P-wave first arrival and are thus extremely difficult to remove without disturbing the P-wave data.

6.1.3 Station F01A (114)

For this station there are three sections to compare: binned and stacked, the former plotted with true relative amplitude and normalised amplitude, the latter normalised only (Figs 6.6, 6.7 and 6.8). On the true amplitude section, little can be seen beyond 220 km. For this reason the stacked true amplitude section has been omitted. In the normalised amplitude plots much more can be seen at larger offsets before the signal disappears into background noise at about 280 km. However the penalty for this is the down weighting of near offset data which becomes less distinct. An example of this is the first arrival up to 60 km which, although clearly visible on the true amplitude section, is barely discernible on the normalised section, being followed by a much higher amplitude arrival. This is a problem of data representation which is overcome by considering both

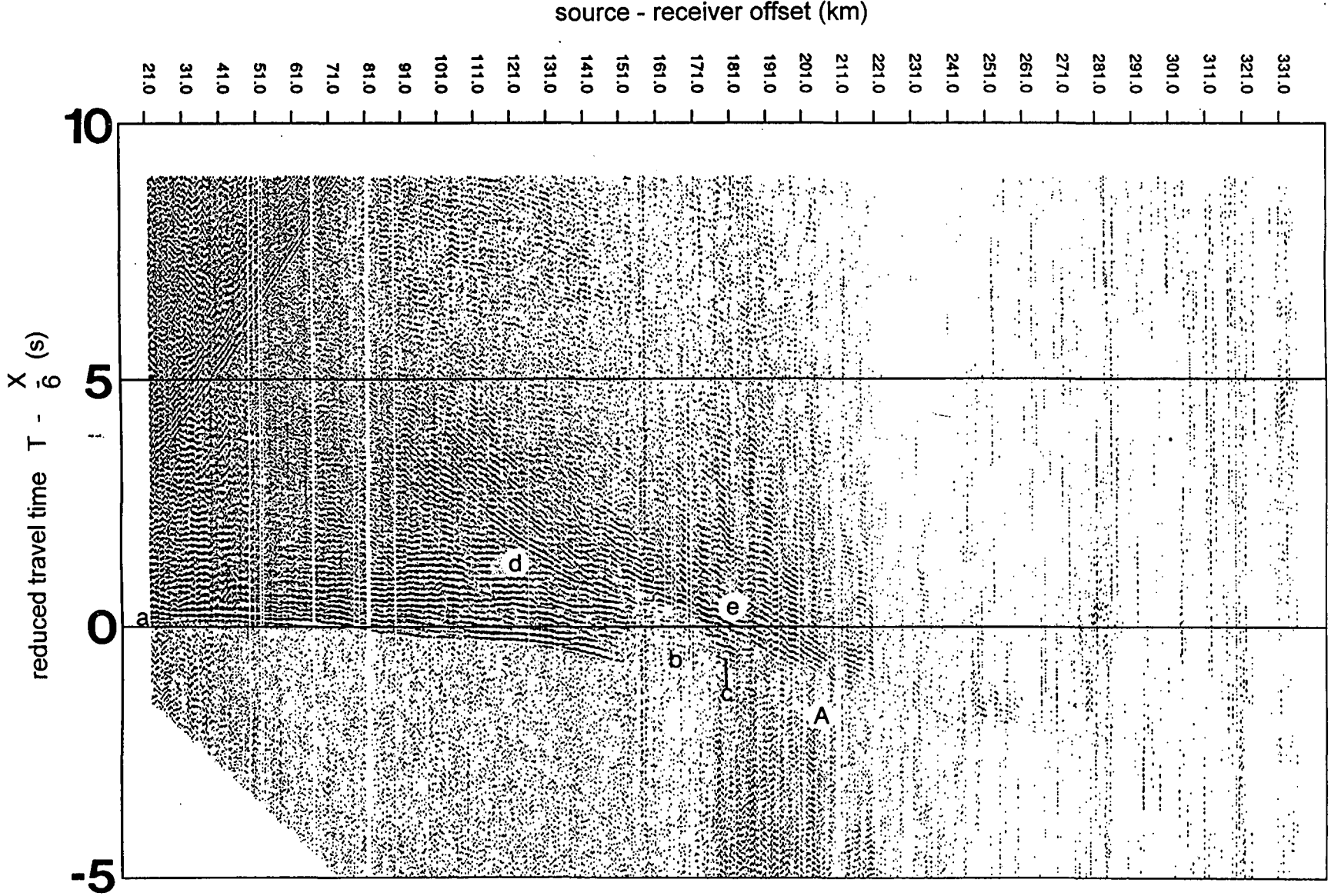


Fig. 6.6 BABEL Line 1, Station F01A(114): binned dataset, true relative amplitude

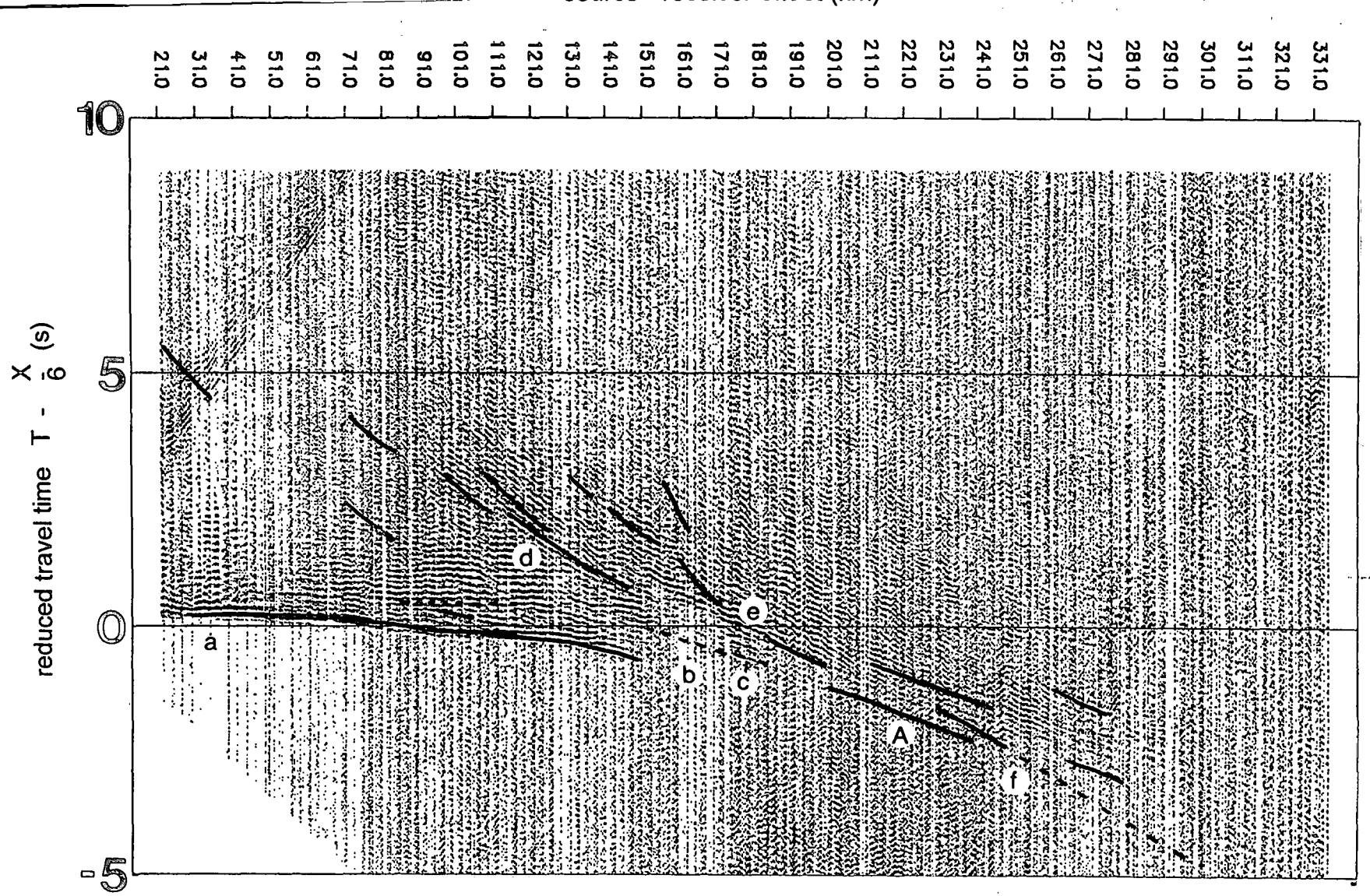


Fig. 6.7 BABEL Line 1, Station F01A(114): binned dataset, normalised amplitude

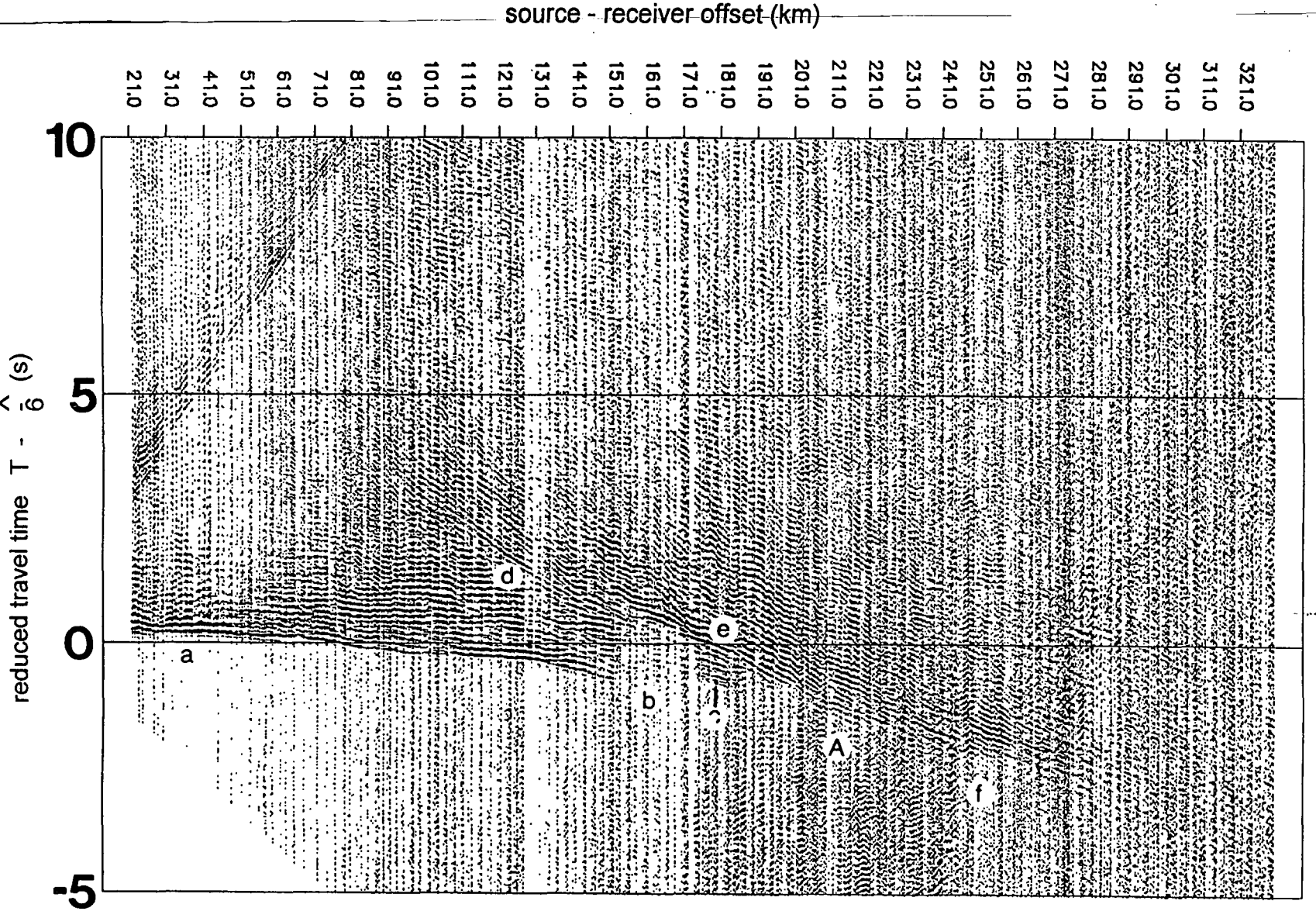


Fig. 6.8 BABEL Line 1, Station F01A(114): stacked dataset, normalised amplitude

types of plot in parallel. The stacked section brings out some of the longer offset arrivals but is otherwise little different from the binned normalised section.

The first arrival appears at the near offset end of the section with a velocity of roughly 6 km s^{-1} (a). There is a large amount of ringing or short-period multiple energy behind this which would seem to obscure any near surface pre critical reflections. This ringing is prevalent over the whole section. The first arrival gradually increases in apparent velocity although the increase is not smooth and the arrival appears broken in places, almost as a series of en-echelon arrivals. At a distance of 130 km the arrival becomes somewhat steeper and then seems to terminate abruptly at 150 km. Although much fainter arrivals continue at a slightly later time, there is a significant low amplitude region between 150 and 175 km (b) in the first arrival which then continues out to 185 km (c) before disappearing again. A steeply dipping curved arrival of high amplitude with the characteristics of a reflection (d) also appears to be truncated at 150 km while a similar one just behind it (e) continues out to greater distances, becoming the first arrival at about 190 km.

Beyond 250 km offset there is faint evidence for a much higher apparent velocity first arrival (f) which also extends nearer to the station as a secondary arrival. Little can be said about the data beyond 260 km as the signal-to-noise ratio is too low.

The dark patch of energy early in the record section between 170 and 250 km is caused by the S-wave from the previous shot. This results in some interference with the first arrivals in the region marked (A) in Fig. 6.7. F-k velocity filtering was applied to remove the S-wave component, as described in Chapter 5. Due to a limit on the maximum size of array

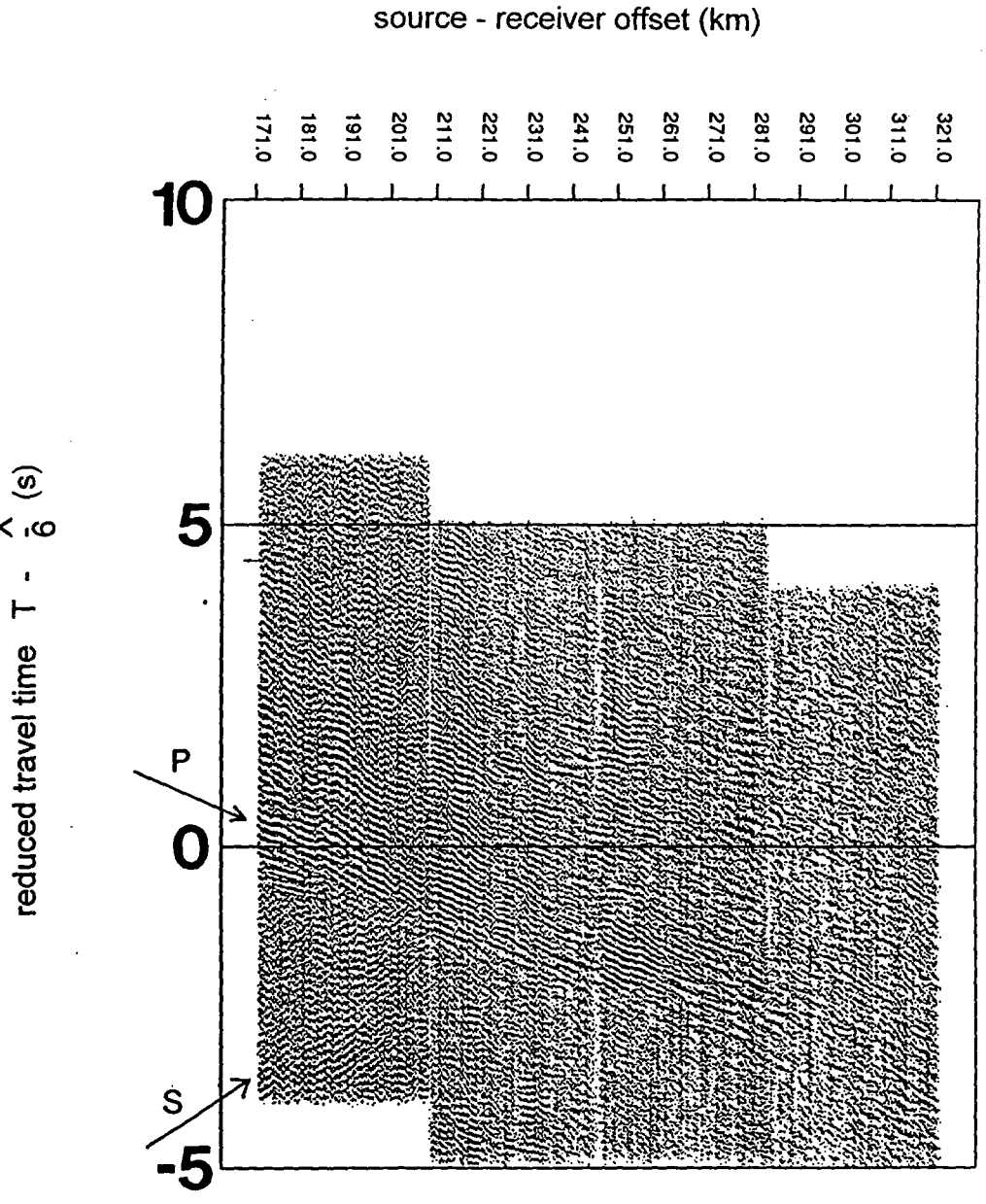


Fig. 6.9 BABEL Line 1, Station F01A(114): far offset data before velocity filtering

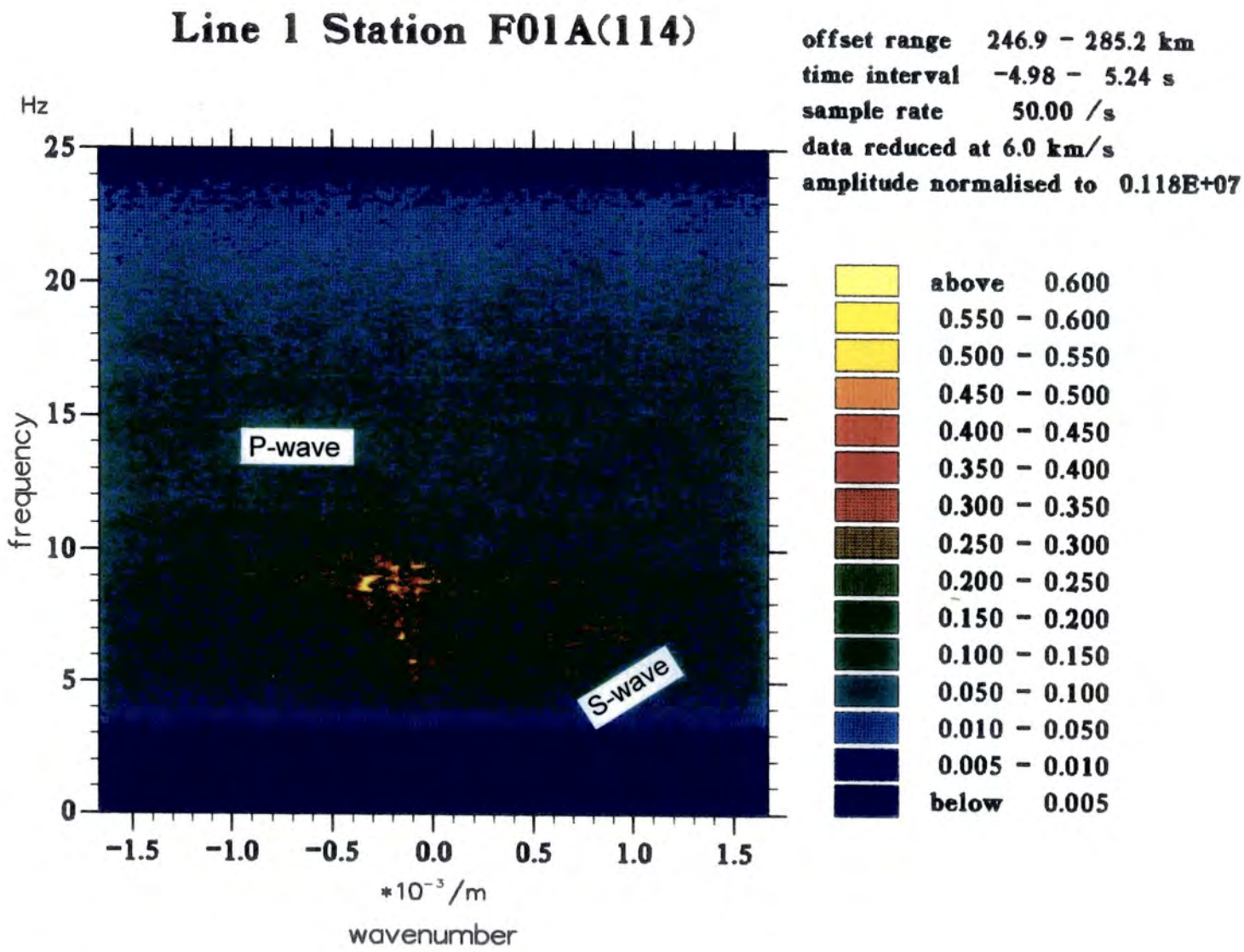
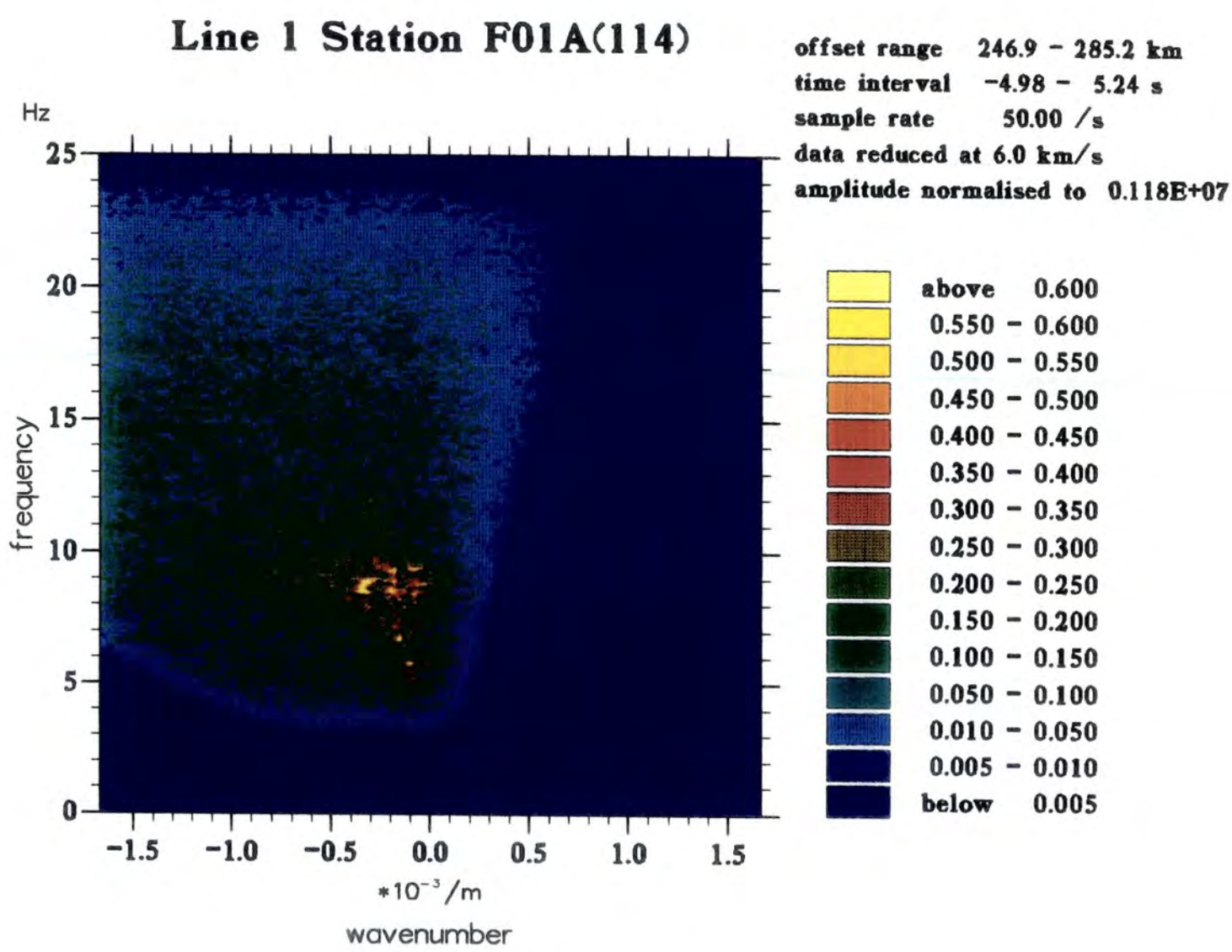


Fig. 6.10 a BABEL Line 1, Station F01A(114): f-k spectrum for part of the data shown in Fig. 6. 9



b) after velocity filtering

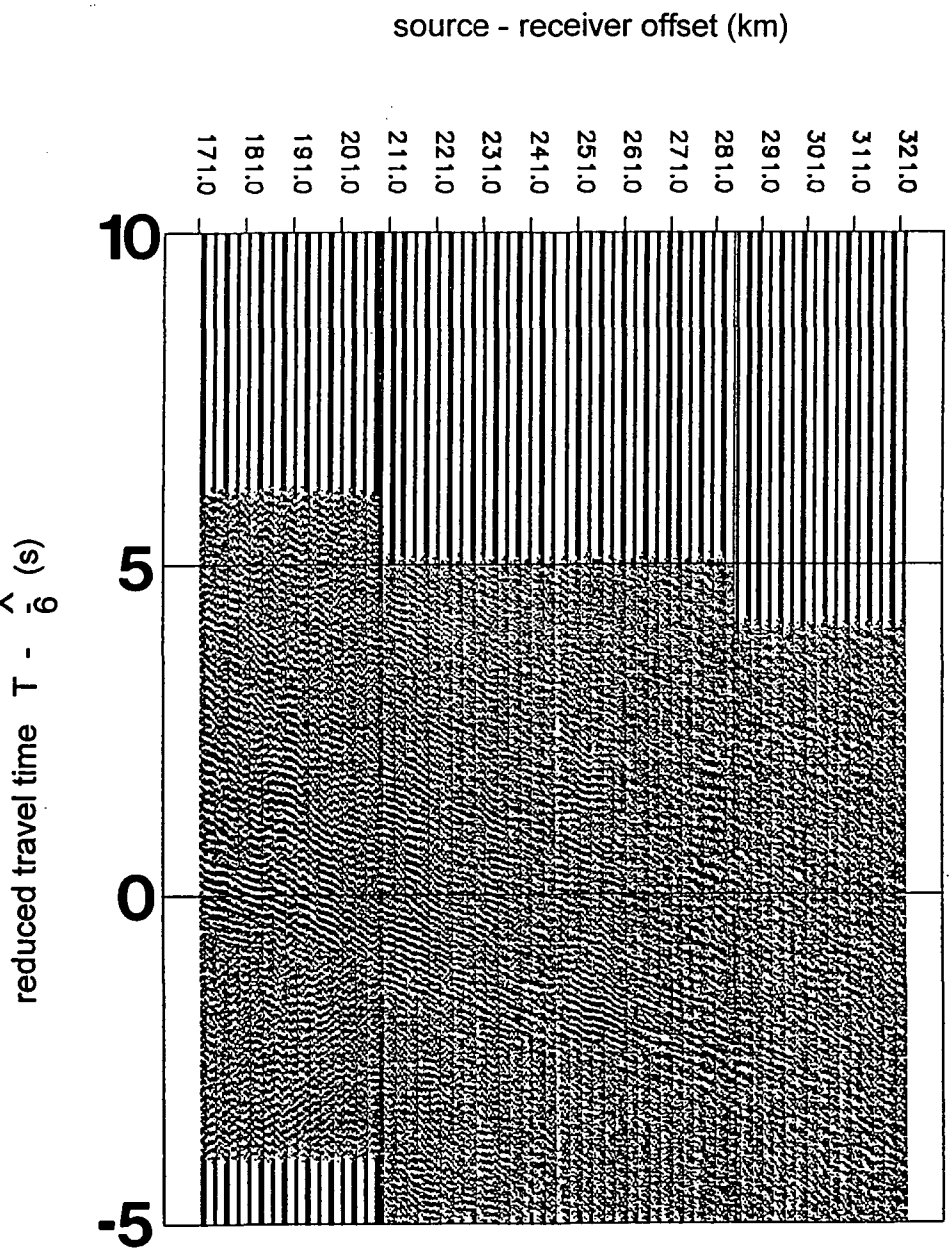


Fig. 6.11 BABEL Line 1, Station F01A(114): the same data as in Fig. 6.9 after velocity filtering

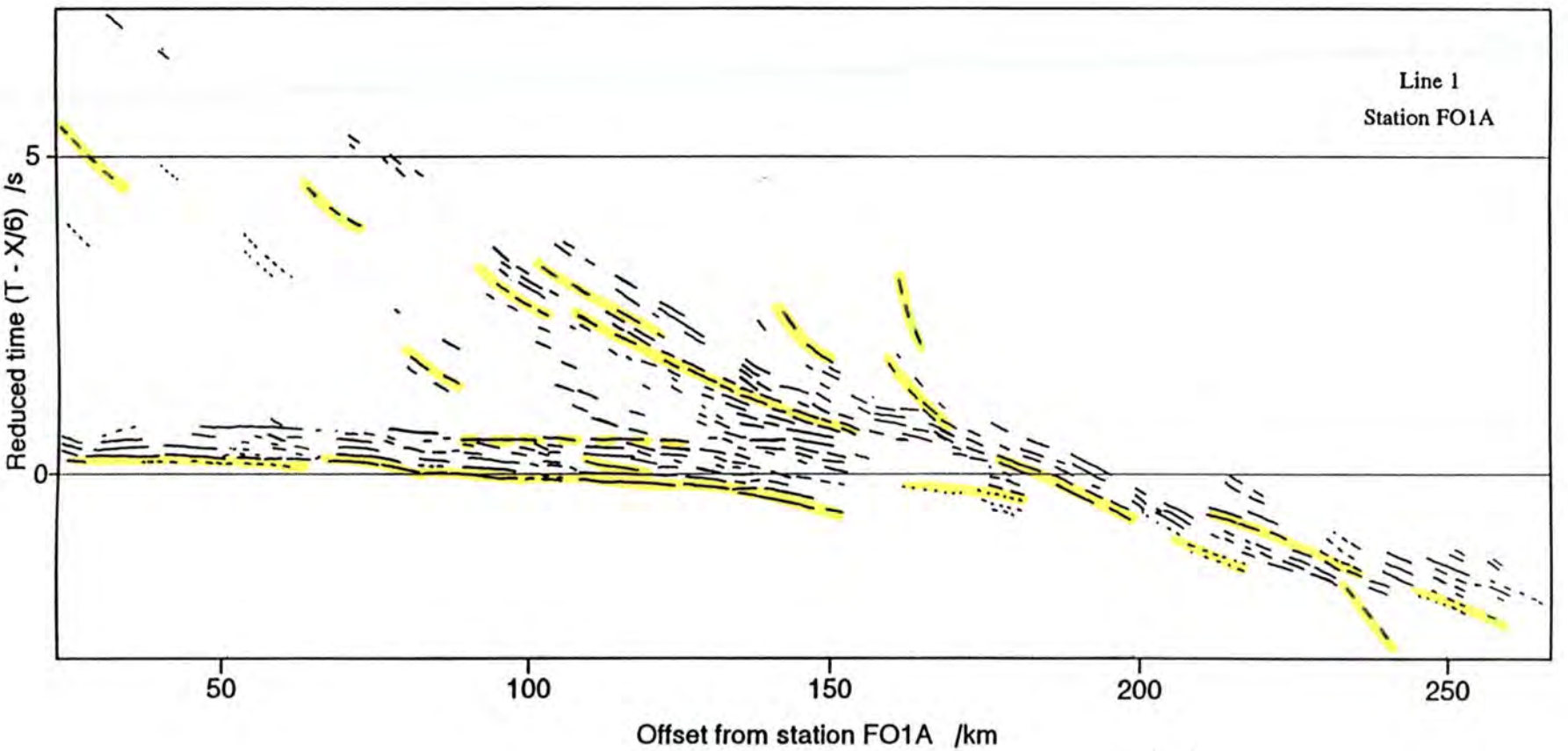


Fig. 6.12 BABEL Line 1, Station F01A(114): line drawing

definable on the computer used, four separate windows of data were taken between 170 and 320 km (Fig. 6.9). It proved difficult to distinguish between the wanted and unwanted data in the f-k spectra, an example which is given in Fig. 6.10, so the results of velocity filtering (Fig. 6.11) were again not very successful and no new P-wave energy was revealed.

The overlay for Fig. 6.7 shows the picks used in modelling and Fig. 6.12 shows a line drawing of the data.

6.1.4 Station 1A

In contrast to the previous station, the data arriving at station 1A was of high relative amplitude to as far out as 320 km so amplitude normalisation was not necessary. However an a.g.c. was applied to the stacked section with a 100 ms window to see whether this would bring out previously hidden arrivals. The true amplitude binned, stacked and a.g.c stacked sections are shown in Figs 6.13, 6.14 and 6.15 respectively. The stacking in this case appreciably increased the signal-to-noise ratio without obviously affecting the data otherwise.

The first arrival (a) appears at the northern end of the section already exceeding 6 km s^{-1} quite considerably. Its amplitude is variable, fading out almost completely before being overtaken by a faster arrival (b) which, as in the F01A data, stops at about 175 km. Close inspection of the sections, especially that to which an a.g.c has been applied, again shows a faint continuation of this arrival to beyond 275 km which is on the threshold of discernibility. A much stronger, approximately parallel arrival (c) comes in about one second later between 185 and 285 km which is itself overtaken by a higher velocity arrival (d). The latter extends northwards

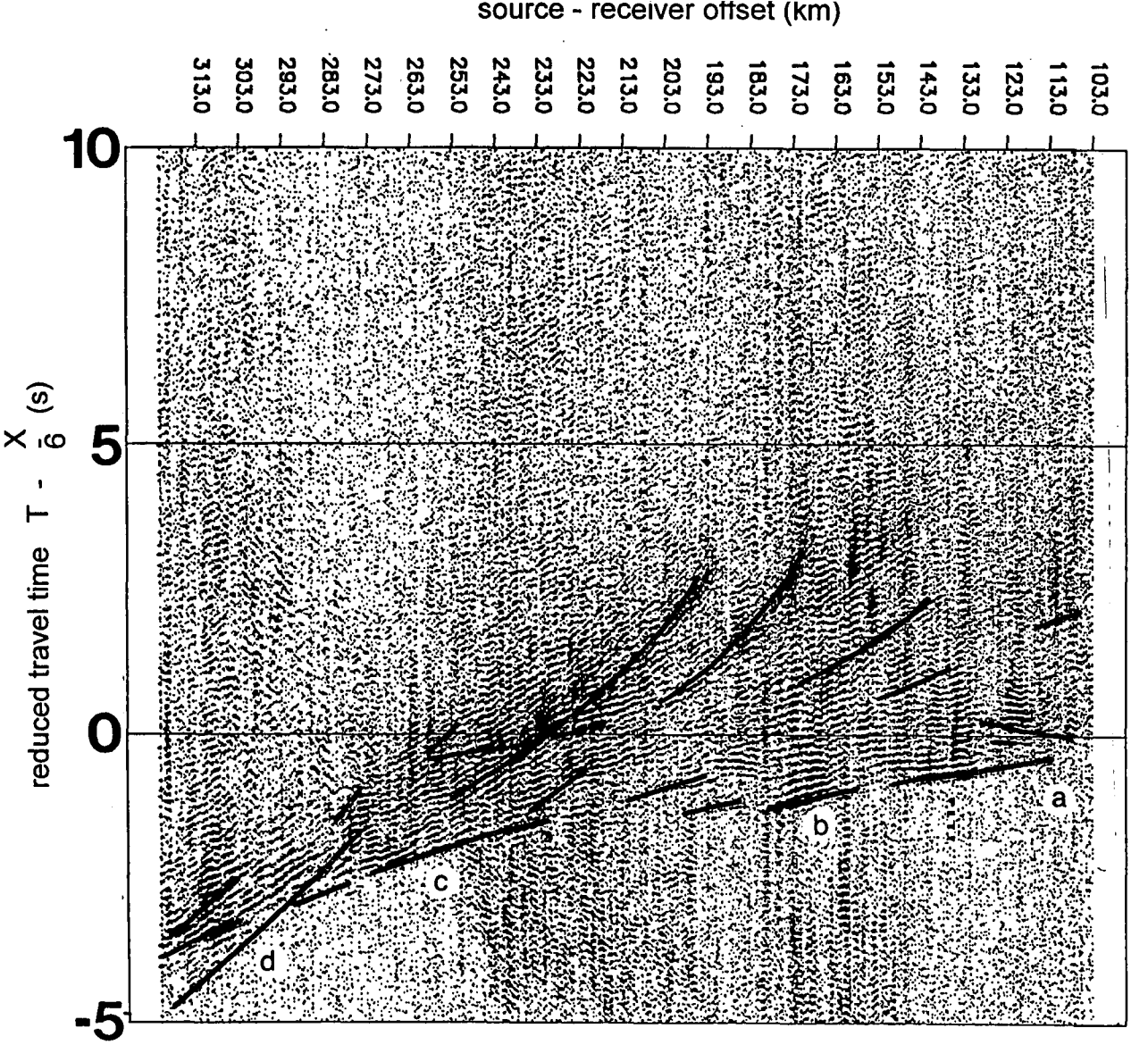


Fig. 6.14 BABEL Line 1, Station 1A: stacked dataset, true amplitude

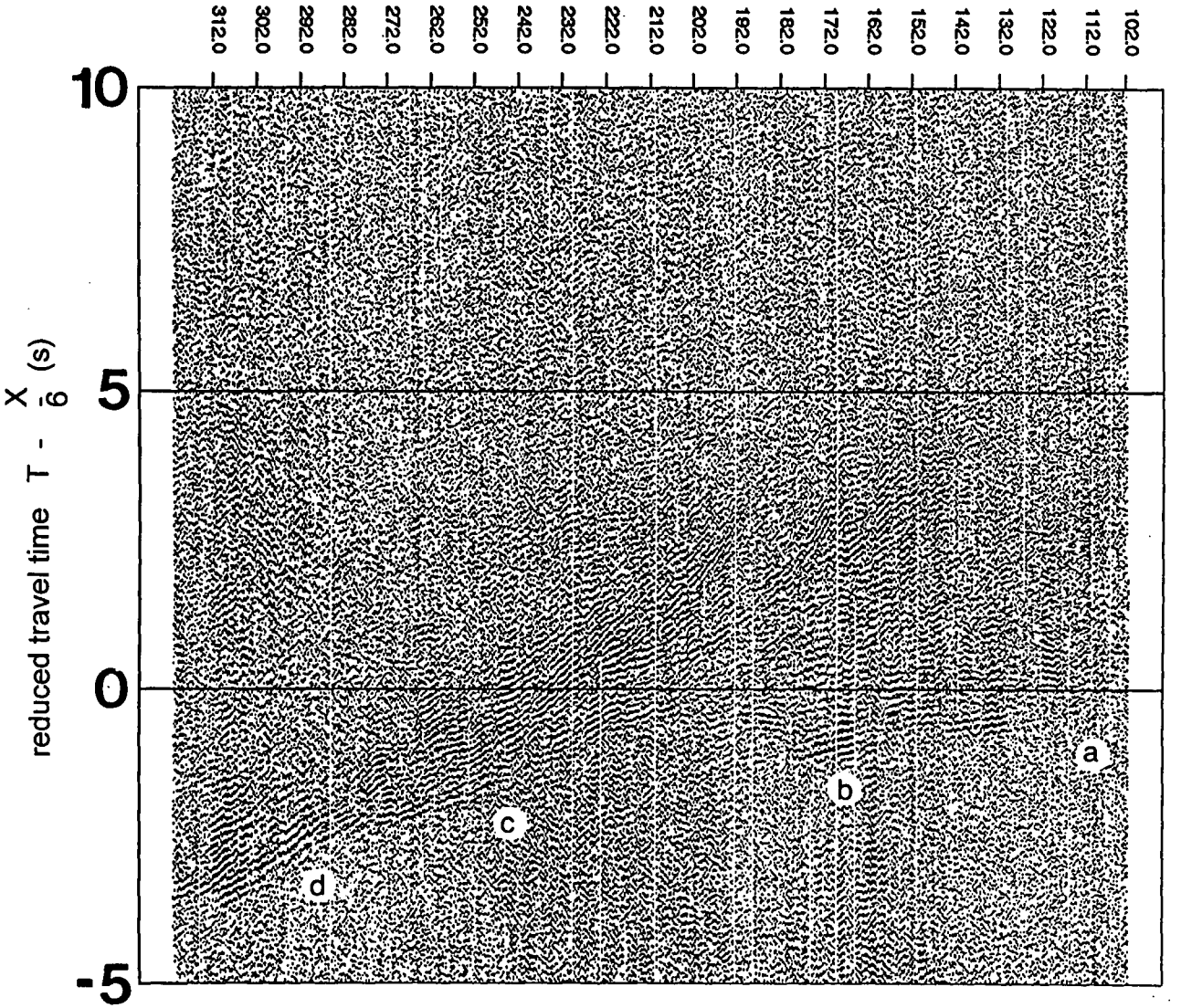


Fig. 6.13 BABEL Line 1, Station 1A: binned dataset, true amplitude

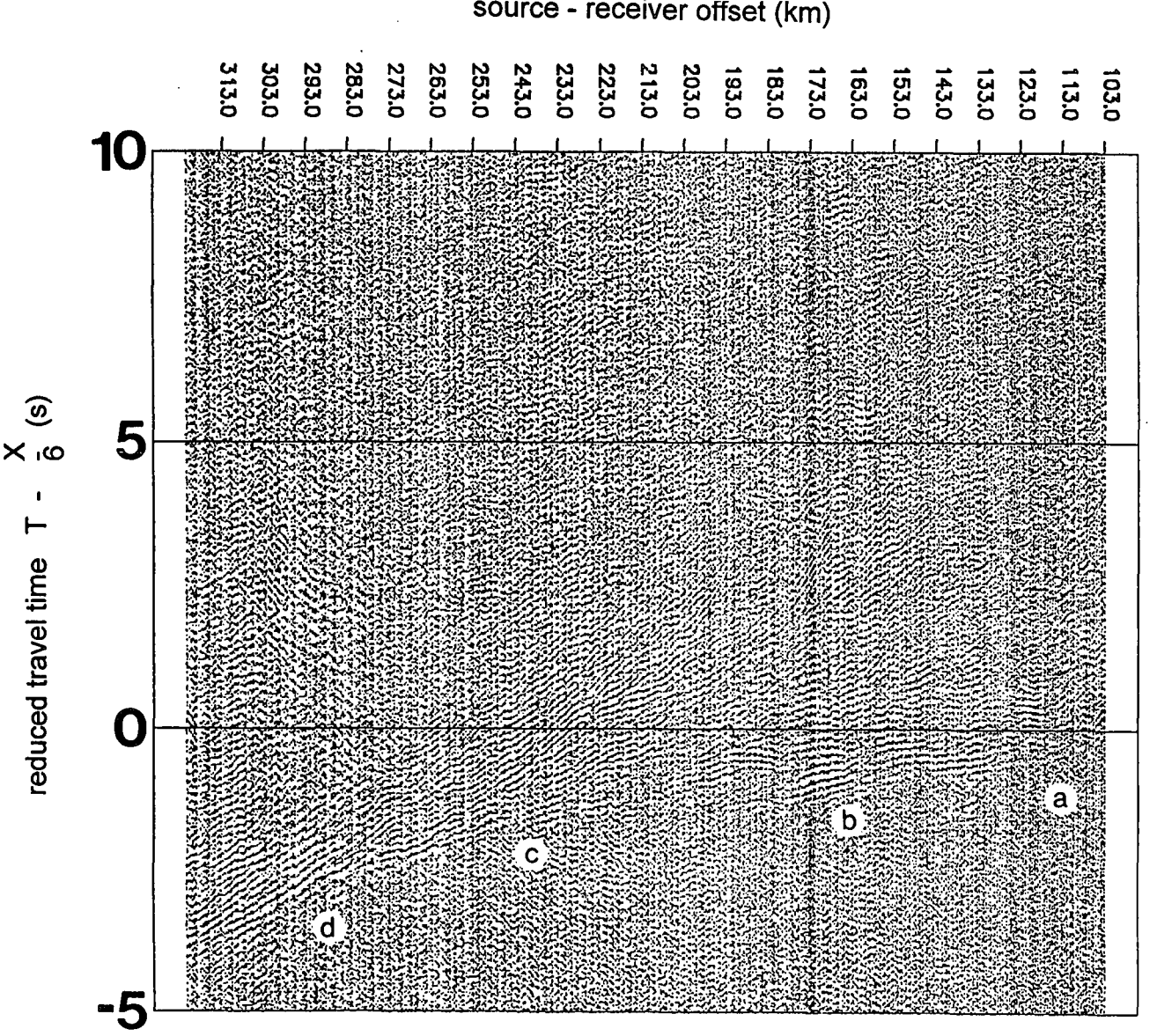


Fig. 6.15 BABEL Line 1, Station 1A: stacked dataset, 100 ms a.g.c

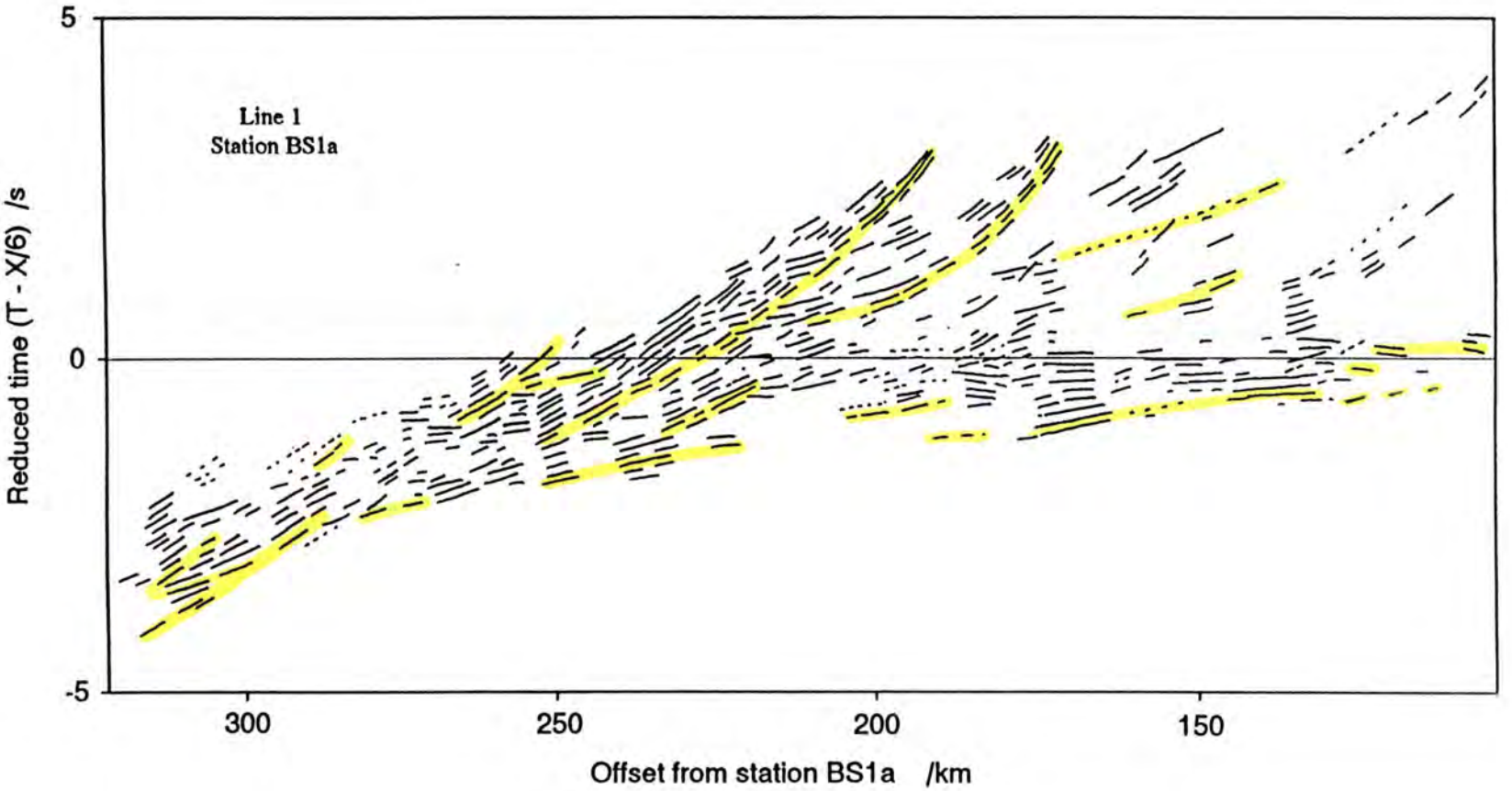


Fig. 6.16 BABEL Line 1, Station 1A: line drawing

across most of the section and appears to fork into several high amplitude, high apparent velocity branches.

The picks and line drawing are shown in the overlay for Fig 6.14 and Fig. 6.16 respectively.

6.1.5 Station 5A

All three varieties of amplitude plot were made for this station but, to avoid needless repetition of data sections, only the three most informative are shown. Fig. 6.17 is the true amplitude binned section, Fig. 6.18 the normalised stacked section and Fig. 6.19 the stacked section with a 100 ms a.g.c. applied.

The first arrival (a) appears to oscillate curiously about an apparent velocity of 6 km s^{-1} to around 170 km. On the a.g.c'd section this looks more like a series of arrivals of differing velocities. A higher velocity arrival is truncated at 185 km just as it becomes the first arrival. Another phase just behind this never actually becomes the first arrival due to the appearance of a much steeper phase at 200 km (b). This becomes indistinct between 250 and 290 km but an even steeper, relatively high amplitude arrival (c) is seen beyond 290 km. F-k velocity filtering once again had little beneficial effect on the region where S-waves from the previous shot interfere.

The highest amplitude on the section is seen in the new first arrival at 200 km (b) and at a steep secondary arrival to the south (d) which would appear to be a continuation of the former were it not displaced slightly in time. There is evidence for the presence of more than one arrival here.

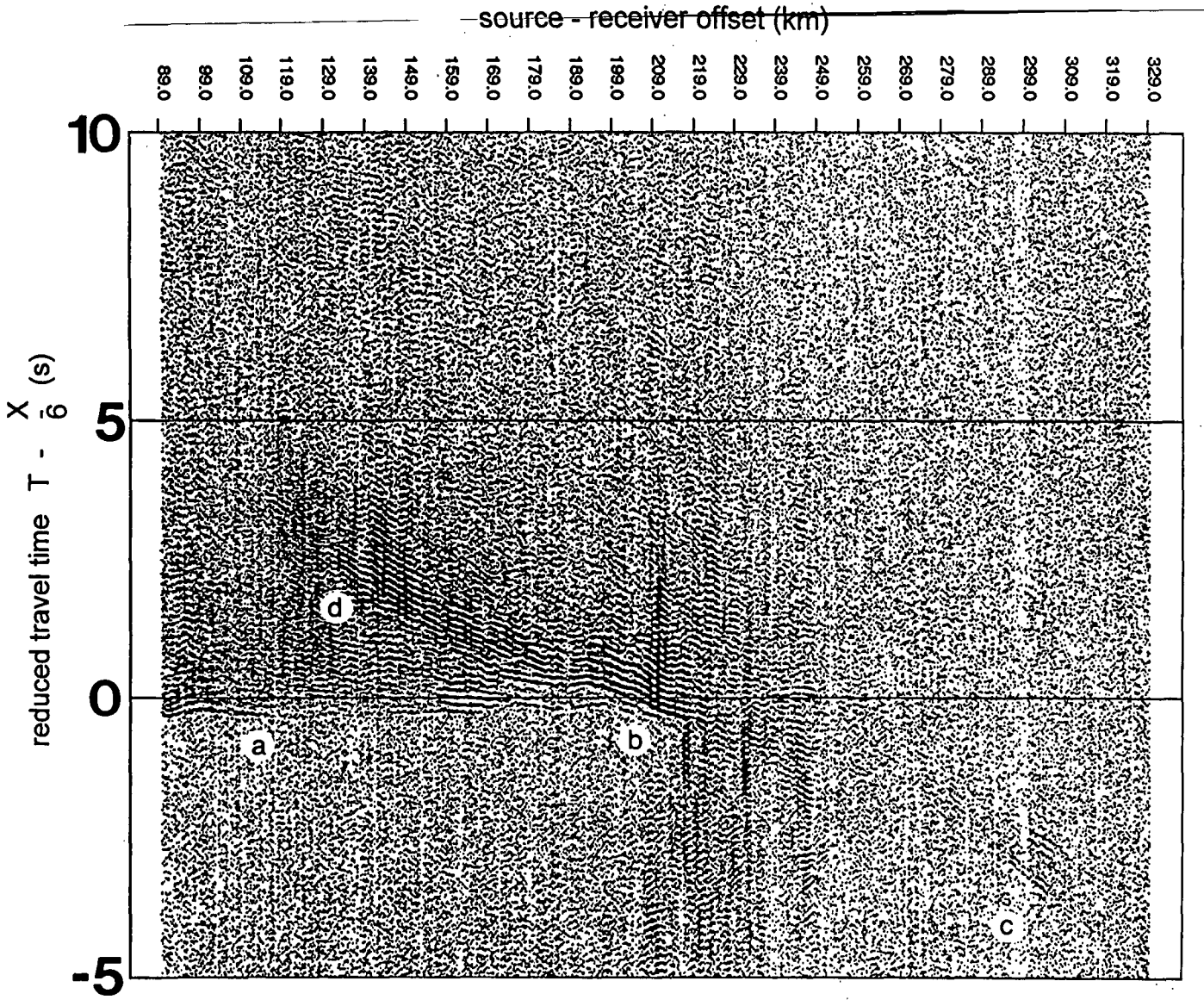


Fig. 6.17 BABEL Line 1, Station 5A: binned dataset, true amplitude

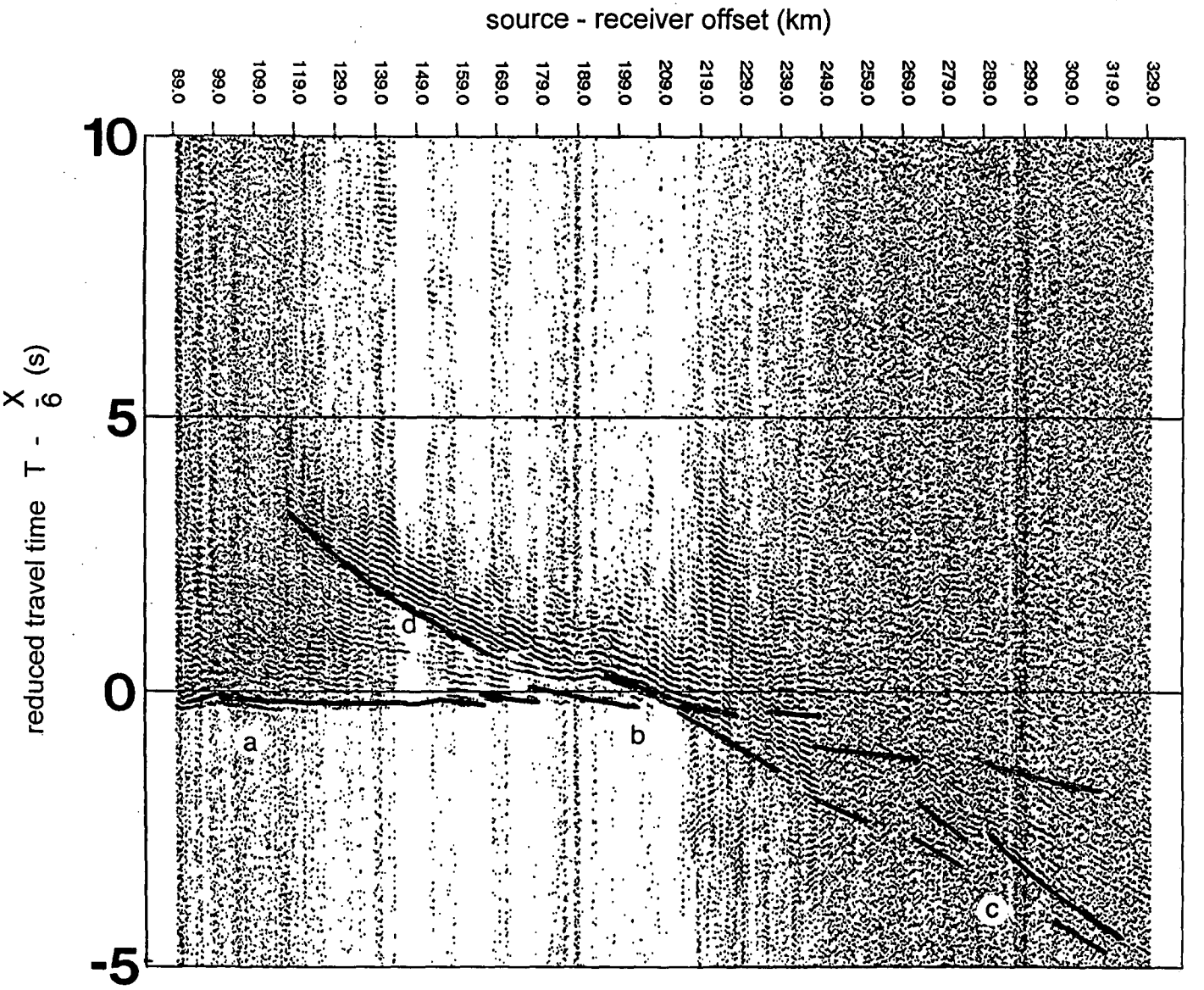


Fig. 6.18 BABEL Line 1, Station 5A: stacked dataset, normalised amplitude

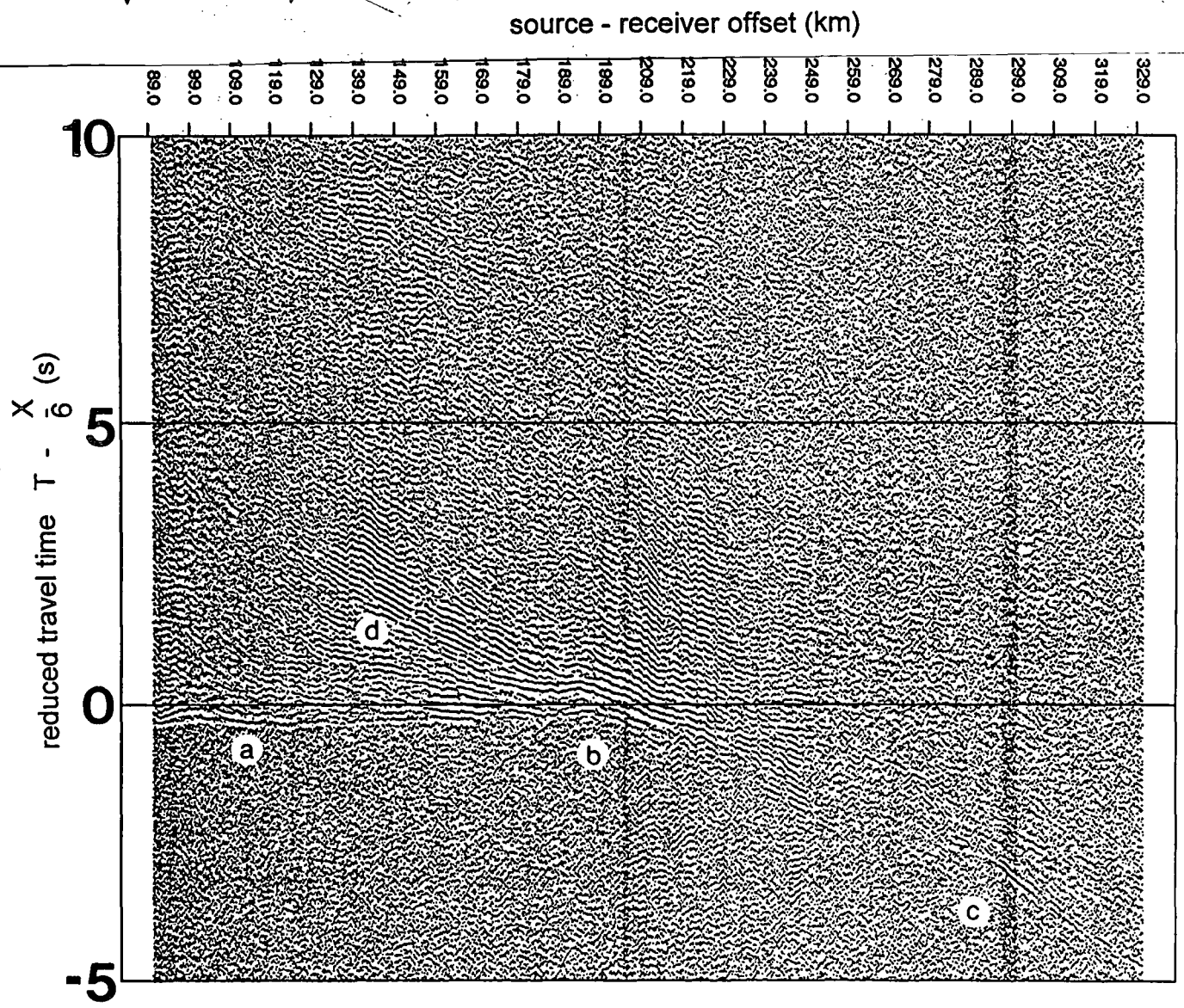


Fig. 6.19 BABEL Line 1, Station 5A: stacked dataset, 100 ms a.g.c.

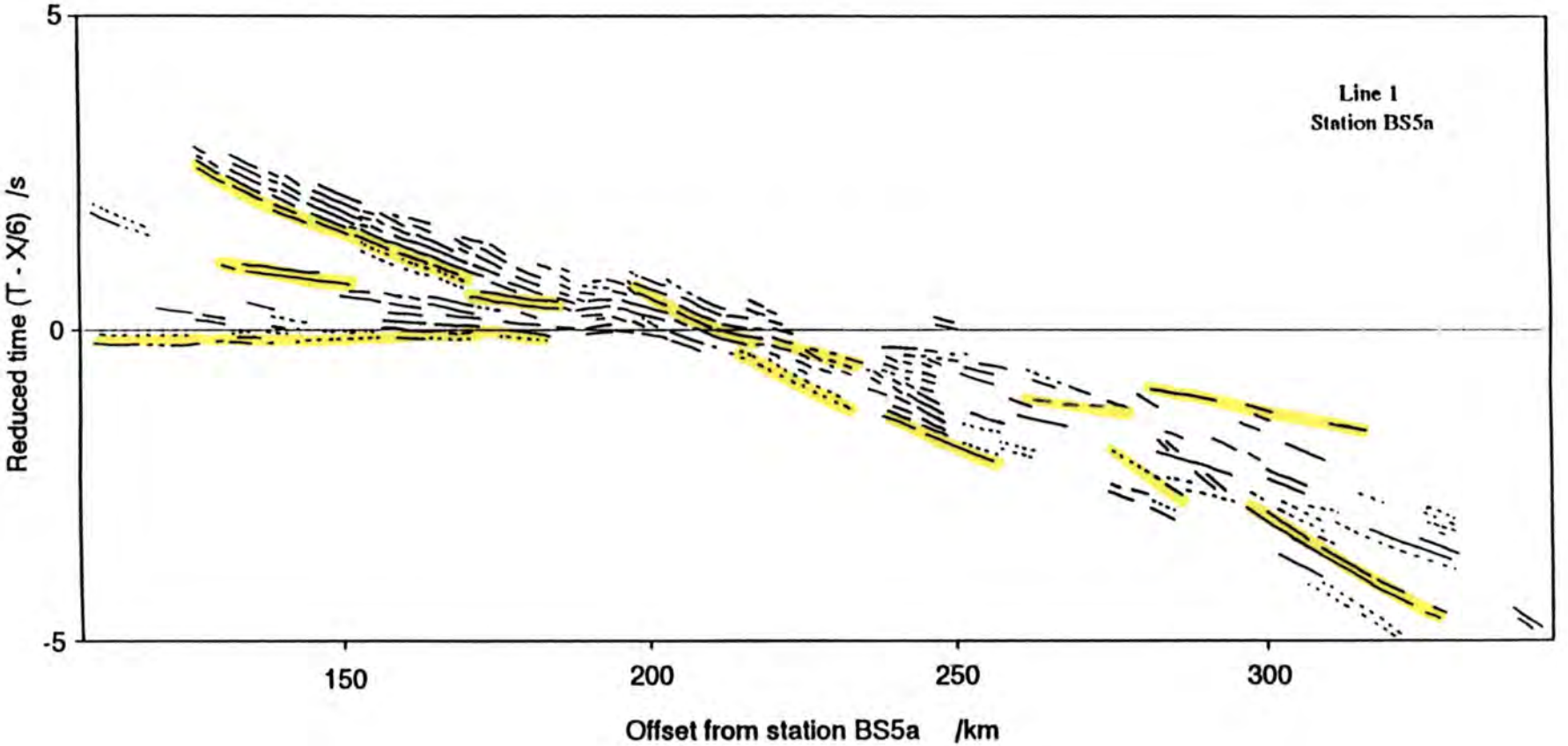


Fig. 6.20 BABEL Line 1, Station 5A: line drawing

The picks and line drawing are shown in the overlay for Fig. 6.18 and Fig. 6.20.

6.1.6 Summary

In general, the stacked sections look clearer, i.e. the signal-to-noise ratio is higher. Stations F01A and 5A exhibit strong attenuation of arrivals from shots further than about 220 km away. For this reason trace amplitude normalisation is required to see data from the whole section. Applying an a.g.c. can in some cases help to show up indistinct arrivals. However any of these types of section must be accompanied by the true relative amplitude binned (i.e. unstacked) plot for a complete interpretation.

In some of the sections, the S-wave from the previous shot interferes with the first arrival between 170 and 290 km. However f-k velocity filtering to remove the S-wave energy did not improve matters, tending instead to smear out the incoherent noise and unwanted signals at the apparent velocities of interest. This is probably explained by the low signal-to-noise ratio in the region and the fact that arrivals from the previous shot are not continuous due to slightly different shot time intervals causing step functions in the x-domain. This should also affect stacking with linear moveout but, in fact, stacking seems to be less affected by this.

In all the datasets, arrivals were found to be made up of short segments with no one phase extending continuously over more than a few tens of kilometres. In addition, amplitudes vary quite dramatically over short distances. This may possibly be explained by the interference of phases of slightly different apparent velocities. Another interesting feature of the data is the large number of high curvature, high apparent velocity

(> 10 km s^{-1}) arrivals which have the characteristic hyperbolic shape of diffractions but are also relatively high in amplitude.

6.2 Normal Incidence Data

Large plots (1.5 x 5.5 m) of the normal incidence data from Line 1 were acquired directly from BIRPS. They were plotted with wiggle traces and variable area fill on a horizontal scale of 50 m/mm (1:50000) and vertical scale of 0.02 s/mm. These were found to be difficult to handle and too large to gain an overall impression. The plots shown here (Figs 6.21 and 6.22) contain every fifth trace and are plotted at a scale of 625 m/mm (1:625000).

A true amplitude plot (Fig. 6.21) shows the general variation in overall reflectivity with travel time and lateral position. However, details of individual reflectors are for the most part difficult to make out. In the southern part of the line, a patchily reflective crust appears to overlie a non-reflective mantle, the base of reflectivity lying between 13.5 and 15 s TWTT. Further north, the reflective upper and lower crust also sandwich a transparent middle crust between them and the reflective crustal base occurs a little later between 15 and 16 s.

A large window a.g.c (3 seconds) allowed more detail to be seen (Fig. 6.22), most of the observed arrivals being upwardly convex and diffraction-like. The majority of these arrivals originate from the middle crust towards the southern end of the line, although the reflectivity at Moho level appears to be also of this type. The Moho is not everywhere clearly defined, especially in the central part of the profile where the decrease in reflectivity appears to be gradational between 10 and 20 seconds. The

a.g.c'd section shows up many dipping events in both directions at the base of the reflective region. These appear to have the hyperbolic shape indicative of diffracted energy.

The upper 4 seconds of the section are fairly featureless, probably due to the source array configuration and processing which was targeted on the lower crust. Exceptions to this are the two prominent reflectors at the very northern end of the profile (boxed in Fig. 6.22). An expanded plot of these is given in Fig. 6.23. A migrated plot of the brute stack (Fig. 6.24) shows there to be, in fact, three features here, all upwardly concave and extending over about 80 km. The characteristic shape, brightness and continuity of these reflectors, and their proximity to outcropping dolerite sills on the Swedish coast, cause them to be interpreted as such (McBride, Snyder and Hobbs, 1992). Similar features are seen in normal incidence profiles from the Siljan region of Sweden and these have been confirmed as dolerite sills by borehole logs (Fig. 6.25, Juhlin, 1990).

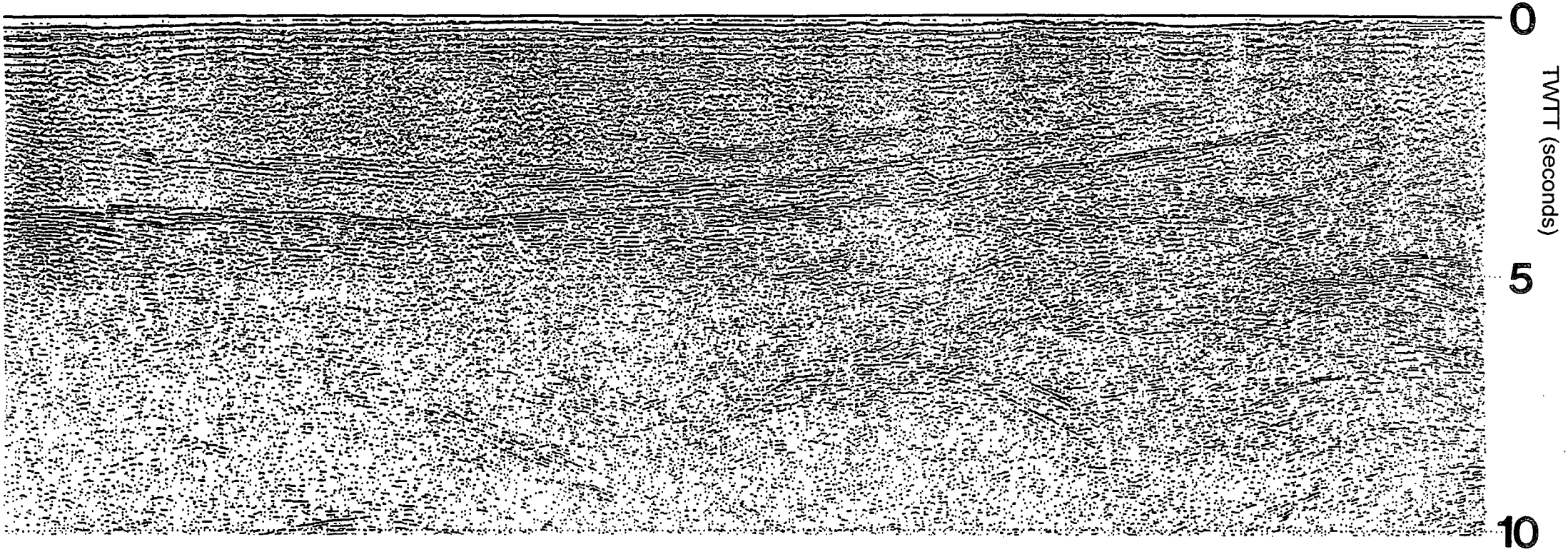


Fig. 6.23 BABEL Line 1, normal incidence: near surface reflectors in the north

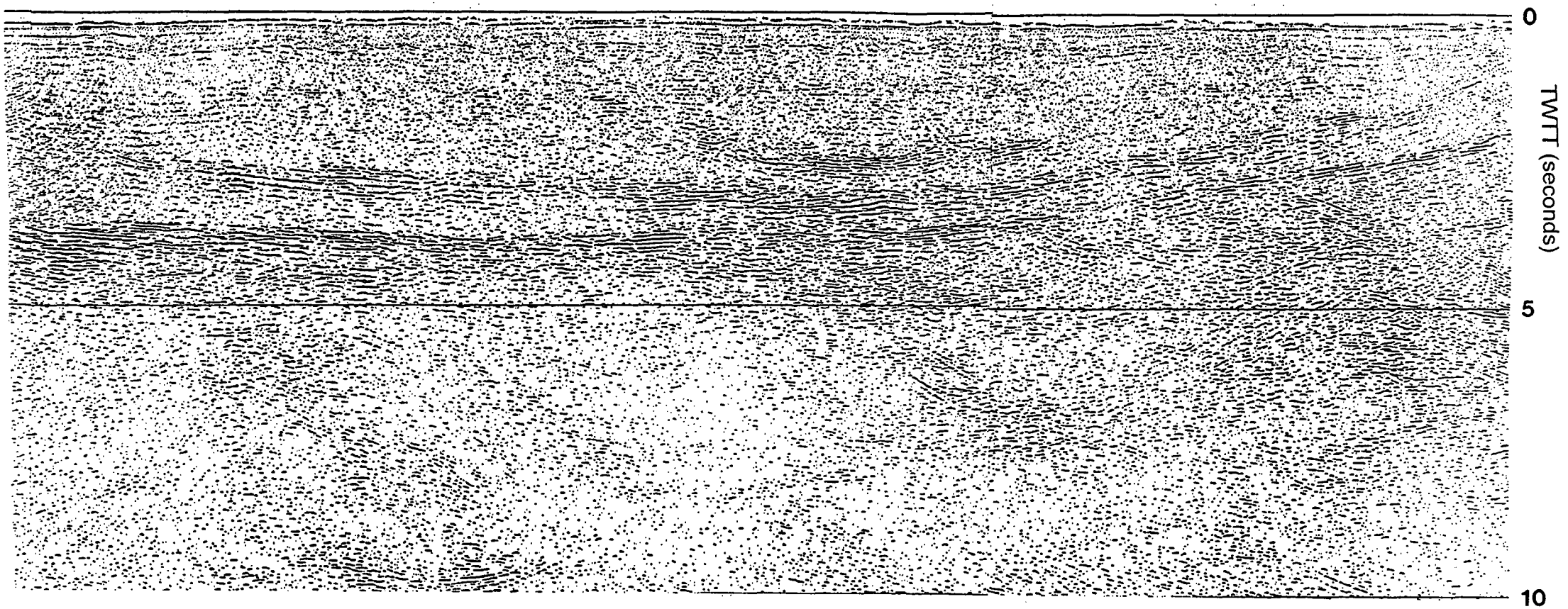


Fig. 6.24 BABEL Line 1, normal incidence: brute stack of near surface reflectors migrated at 6.5 km s^{-1}

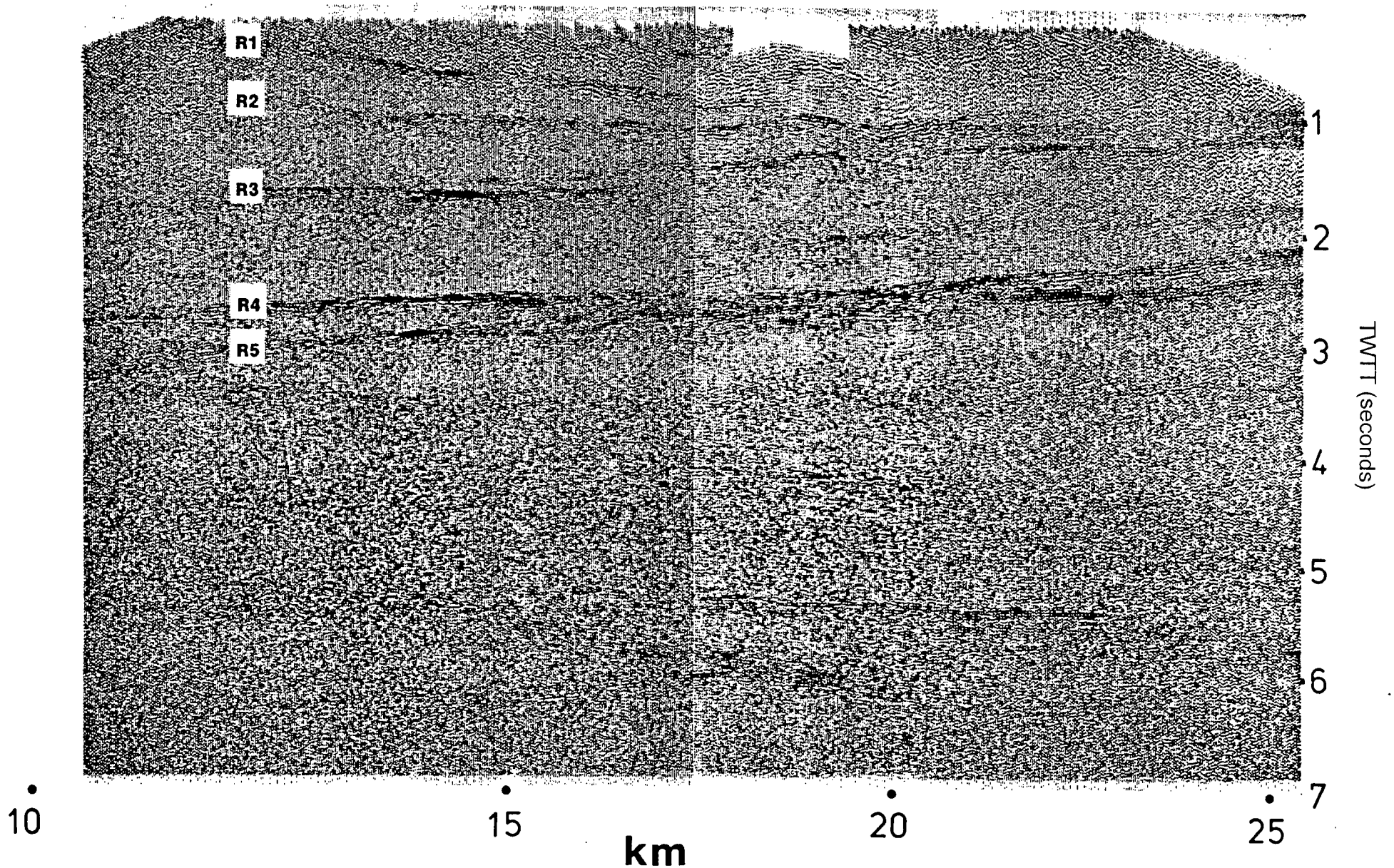


Fig. 6.25 Dolerite sills observed in normal incidence profiles near the Gravberg1 borehole in the Siljan Ring in south central Sweden (Juhlin, 1990)

7. Modelling

7.1 Introduction

The seismic modelling package used in this study was BEAM87 written by Dr. V. Cerveny at the University of California, Berkeley (Cerveny, 1985b). It uses the Gaussian beam method to calculate synthetic seismograms, a modification of asymptotic ray theory. A 2D gravity modelling program written by Prof. M. H. P. Bott at the University of Durham was also used to constrain the in-line seismic model.

Several techniques are available for the modelling of wide angle seismic data and the calculation of synthetic seismograms, of which the reflectivity method, finite difference and finite element methods, and asymptotic ray theory (A.R.T) are the most well known. These are compared by a number of authors (e.g. Fuchs and Müller, 1971, Kennett and Harding, 1985, Weber, 1988). The only exact method is the reflectivity method; all the others are approximations which introduce their own errors. However, the application of the reflectivity method is limited to laterally homogeneous media and it is therefore unsuitable for modelling where strong lateral inhomogeneities are expected. Of the other methods, the finite difference and finite element methods give the closest approximations provided the grid is sufficiently fine to avoid artifacts. They are very powerful, yielding accurate seismograms for complicated media. Their major drawbacks are the vast amount of computing power required to use them with a fine grid, and the difficulty in interpreting the highly complex resultant seismograms (Weber, 1988). Ray theory, despite being less

exact, is simpler and faster, making much smaller demands on computational resources.

In the ray theory approach, employed by BEAM87's predecessor SEIS83, the wavefield is broken down into a number of elementary waves based on the path taken through the model. The trajectory and travel time of a series of rays are calculated for each elementary wave (Cerveny, 1985a). These rays are traced through the model from a starting point to specified endpoints by varying the initial take-off angle. This is called two-point ray tracing (other methods of specifying the rays may also be used). Amplitudes are modified by transmission and reflection at interfaces and corrected for geometric spreading. Summation of all the elementary seismograms obtained in this way yields the final synthetic section. The method is a high frequency approximation, based on the assumption that the seismic wavelength is shorter than the dimensions of the structures in the model. Thus the velocity field must not vary too rapidly. As the method does not give a full solution for the complete wavefield, the theory also breaks down where the ray field is not regular, such as in the critical region and caustics, and edge diffractions cannot be calculated. The technique's advantages lie in its flexibility in modelling 2D laterally inhomogeneous media and in its speed.

Some of the drawbacks of A.R.T can be overcome by modifications and extensions to the theory. The most well known of these are the WKBJ-Maslov method (Chapman, 1985) and the Gaussian beam method (Cerveny 1983, 1985b, Weber 1988). The latter was used in modelling BABEL Line 1, implemented by V. Cerveny in the BEAM87 package.

The Gaussian beam method uses beams of seismic energy of finite width with bell-shaped amplitude cross-section instead of the infinitesimally

narrow rays considered above. The wave equation is solved close to the central ray which removes the singularities apparent in the ray method.

Ray tracing is carried out as before and ray amplitudes determined, without the geometric spreading correction. The contributions of the Gaussian beams are then evaluated, the beam width and phase front curvature being modified at each model interface. The receiver response is calculated by superposition of all the beams arriving in the vicinity of the receiver.

The method is more stable than the pure ray theory in singular regions such as caustics and critical regions and it produces more reliable amplitudes whilst retaining the advantages of speed and flexibility. It is also less sensitive to minor details in the model (Cerveny 1985b). However the effects of surface curvature are not taken into account in the treatment of reflections and transmissions and, although diffractions may be represented, amplitude and phase information are not accurate (Kennett and Harding, 1985).

The BEAM87 package itself, in the form used, consists of four main programs with an additional one written at Durham by P. A. Matthews to convert the synthetic data to the same format as the real data. Fig 7.1 shows a flow diagram of the modelling process. The model is defined by a number of interfaces, each specified by up to 30 points and interpolated by a cubic spline. The interfaces also represent isovelocity lines, velocities being specified immediately above and below each interface. A vertical linear interpolation is applied within each layer to evaluate the velocity field. Two other options are included in the package which involve either a piece wise bilinear interpolation or a cubic spline applied to a grid of velocity specifications within each layer. In these last two methods, the interfaces

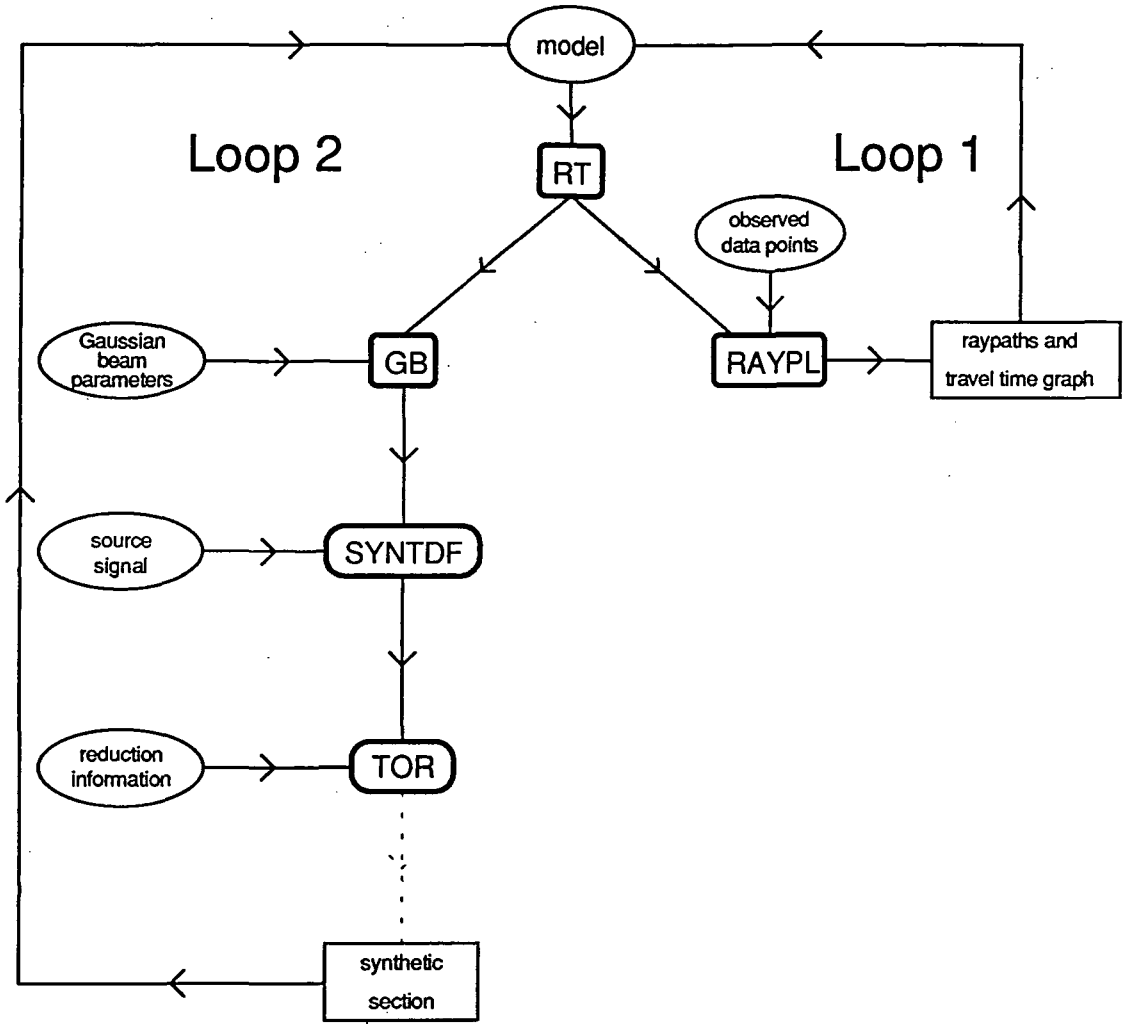


Fig. 7.1 Flow diagram of the modelling process

are not velocity isolines and have the advantage that lateral velocity variations can be represented within a layer. However the bicubic spline produces oscillations in the velocity field even after smoothing and the bilinear interpolation method can give unstable results (Lewis, 1986, West, 1990). Thus, as the most reliable method, the linear interpolation procedure was the only one implemented in the version of BEAM87 used.

The main program RT performs the raytracing between the source and a specified set of receivers along the Earth's surface. Travel times, ray paths and amplitudes are evaluated for each elementary wave. The amplitudes are reduced to have unit vertical amplitude at the source. The travel times, ray paths and amplitudes can then be plotted by the program RAYPL (modified by C. Prescott and P. A. Matthews to use the UNIRAS graphics library) along with the picked arrivals which are digitised and included in the control file. This is generally the first stage in modelling and is repeated iteratively, adjusting the model each time to get a better fit to the digitised travel time data.

In the second stage, GB calculates the frequency response at the receivers for each elementary wave by the summation of Gaussian beams, taking into account source radiation pattern and dissipation effects. The method depends on having a high density of rays near the receiver as the response is calculated by a weighted summation of several overlapping beams.

Seismograms are produced by SYNTDF with a user-specified source signature. The seismogram at each receiver is calculated by combining the frequency responses for all the elementary waves to give one frequency response. This is multiplied by the Fourier transform of the source signal, and the inverse Fourier transform gives the synthetic

seismogram. The far field source signature supplied by Prakla Seismos was used as the input signal for Line 1 synthetics. A double cosine filter is included in the program, mainly to filter out low frequencies as the Gaussian beam method is based on a high frequency approximation. The limits chosen for this were similar to those used in band-pass filtering the real data. The program was adapted from the original to output the synthetic data in ISEGY format. This could then be reduced and written out in the same way as the real data by TOR, written by P. A. Matthews, and plotted out.

This second modelling stage can also be repeated iteratively to match amplitudes as far as possible. It must be noted, however, that the synthetic trace amplitudes are normalised so that the maximum amplitude in each trace = 1. The section as a whole cannot therefore be compared directly with true amplitude data sections.

One must also be aware that BEAM87 is designed to model one shotpoint with many receivers while for the BABEL data we have many shotpoints and one receiver. Whilst raypaths are totally reversible, amplitudes are not and another source of error is introduced. Fortunately, the difference is very small unless P to S converted phases are being modelled (Matthews, 1993).

The programs in the package were also adapted by P. A. Matthews to model normal incidence data.

7.2 Preparation of data for modelling

Before the first stage of modelling could be carried out, the picks of seismic arrivals on the data sections had to be digitised. This was done semi-automatically on a digitising table attached to a SUN workstation. A deskewing transformation was applied to remove the effect of misalignment of the paper section on the table. The input file to RAYPL requires that the data are in unreduced form so a further operation was performed on the digitised data points to dereduce them. The program TRANSFORM performed both the deskewing and dereduction. It was written by the author and can be found in Appendix 1.

In order to model a reversed profile, it is necessary to locate the receivers on the model and modify the x-coordinates of the picked data points to maintain the correct shot-receiver separations. Stations F01A and 101P are near enough to the strike of BABEL Line 1 to be considered as in-line (Fig. 7.2a). For this reason, F01A was taken as the horizontal origin of the model. F01A and 101P are separated by 358.4 km so the transformation applied to the x-coordinates of the data points for 101P is:

$$x' = 358.4 - x$$ where x' = new coordinate, x = old coordinate

With the off-line stations 1A and 5A, the situation was three dimensional. A problem therefore arose in how to model the data with a 2D modelling package. In order to get around this problem, a transformation was applied to the station and shotpoint coordinates so that they all lay along the model profile. To do this, the station position had to be projected on to the line of strike of the profile whilst maintaining the shot-receiver

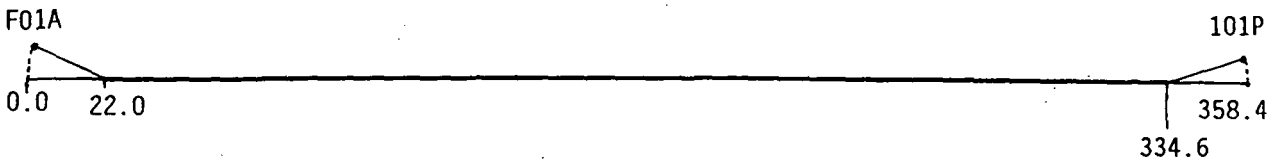
offset. The approximation to in-line stations is most accurate at the greatest offset where the angle between the raypaths and the line is at a minimum. The furthest shotpoint was therefore held stationary during the transformation of the shot and station coordinates.

Firstly the shotpoints were rotated in the horizontal plane about the receiver so that they lay on a straight line between the receiver and the furthest shotpoint. This has the effect of making the shotpoints appear closer together but shot-receiver ranges are unaffected. The whole line was then rotated again about the furthest shotpoint so that it was colinear with the model profile. Figs. 7.2b and c show the geometry of the two stages of this operation for the two off-line stations. For convenience, the same model coordinate system as for the in-line stations was used, i.e. the origin was taken as the location of station F01A. The mathematical operations applied to the two sets of data points are as follows:

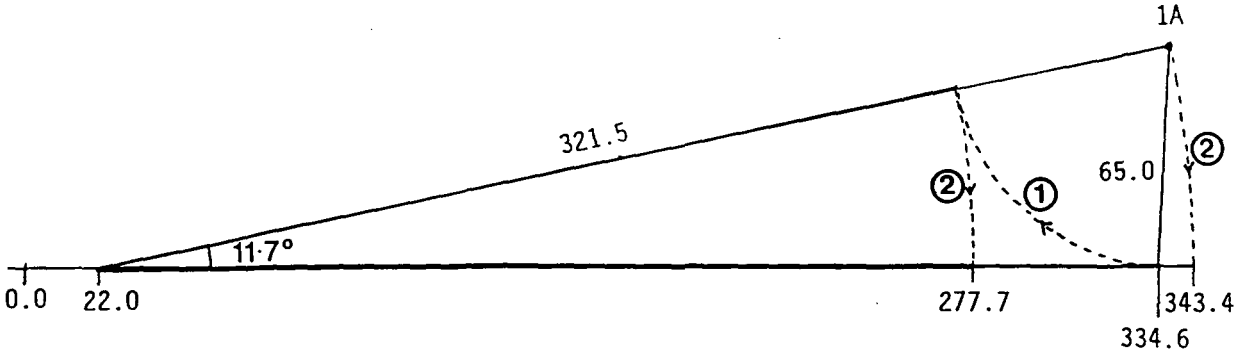
1A:	$x' = 343.4 - x$	where x = shot-receiver offset
5A:	$x' = x - 24.0$	x'= model coordinate

Obviously the shotpoints were dislocated from their true locations with respect to the in-line model, only the furthest shotpoint in each case retaining its original position. This fact must be remembered when comparing and interpreting the models.

a) stations 101P and F01A



b) station 1A



c) station 5A

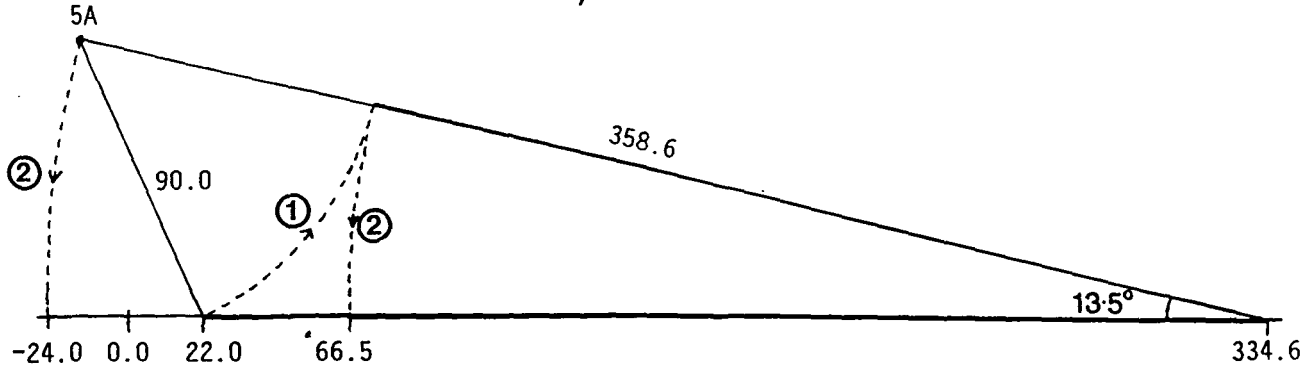


Fig. 7.2 Relationship of Line 1 and wide angle stations to the model coordinate system

7.3 The Models

7.3.1 Introduction

The first dataset to be modelled was that recorded at station 101P. After producing an unreversed model for the 101P data, the F01A data was also modelled independently. These two models were then combined, modelling alternately from each end of the line until a good fit had been obtained to both datasets. Several revisions of the model were made, mainly in an attempt to curtail arrivals produced by the model which were not seen in the data and to match amplitudes (in a qualitative rather than quantitative sense). Two dimensional gravity modelling was then carried out based on the seismic model and gravity data published by the BABEL Working Group (1993). The seismic model was further revised to comply with the constraints of the gravity modelling.

A feature of the data is the complexity of the arrivals with several arrivals of differing apparent velocity, sometimes quite high, often being superposed. Large variations in amplitude over short distances are also common. It was postulated during the modelling process that these features may result from the interaction of the seismic energy with an undulating boundary. To investigate this further, a series of hypothetical models were created with undulations of various wavelengths on the Moho. Fig. 7.3 shows the basic model with no undulations. It consists of a simple three layer crust 50 km thick. Diving rays in the lower crust and upper mantle are shown, along with the Moho reflection. Below this is the synthetic seismogram generated from the model with the indicated raypaths using the source wavelet given in Fig. 4.6. The synthetic section

shows a certain amount of ringing. This is an artifact of the synthetic generation program produced by filtering of the wavelet. Figs. 7.4 to 7.7 show the effect of different wavelength Moho undulations on the seismic record. For the longer wavelengths, the position of the receiver relative to the peaks and troughs of the undulation has a strong influence on the outcome (Figs. 7.4 and 7.5).

Obviously for the shorter wavelengths, the amplitude of the oscillation is unreasonable and merely serves to emphasise the effect. However, isolated offsets in the Moho modelled in previous surveys in the Baltic Shield do have comparable magnitude and boundary curvature (see Chapter 3).

In the last of these models (Fig. 7.7), not enough points were available in the interface definition for the oscillations to extend across the whole model. From the undulating part of the boundary it is clear that a highly complex set of individual arrivals are merged into one broad band of energy, even when the ringing artifacts are discounted. A similar effect has been created by Gibson and Levander (1988) with a mathematical model generated by imposing small 2D random velocity variations (or scatterers) on a uniform vertical velocity gradient and using a finite difference technique to obtain a synthetic seismogram.

From these hypothetical models it can be concluded that structures and irregularities on a seismic boundary have a large effect on the seismic record. It is possible for a small section of an interface with high curvature to produce an arrival over a large distance range. The limiting case of this would be a diffraction.

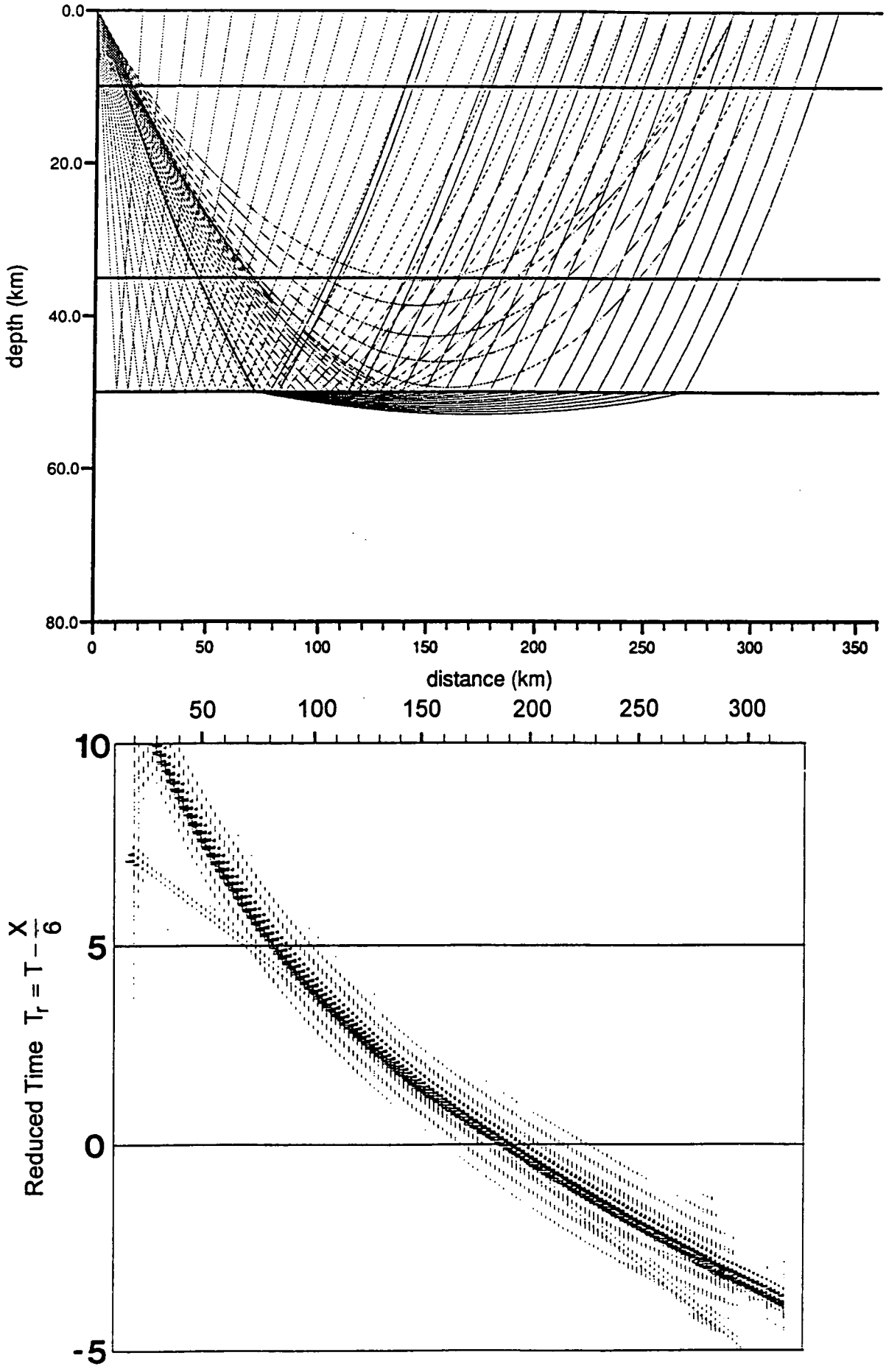


Fig. 7.3 Lower- and sub-crustal raypaths and corresponding synthetic section for a simple three-layer crustal model with no lateral variations

a) **Long wavelength undulations on the Moho**

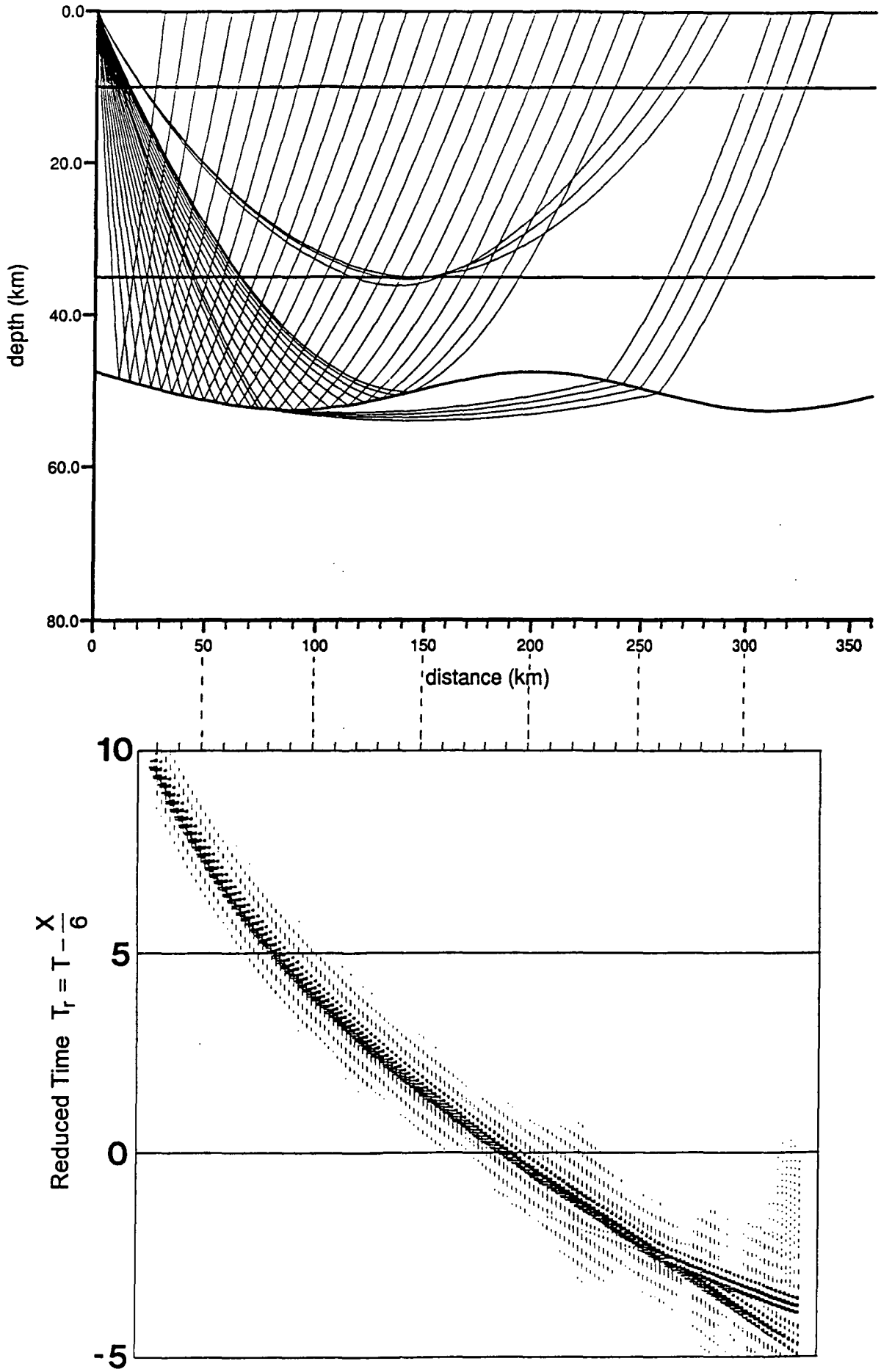
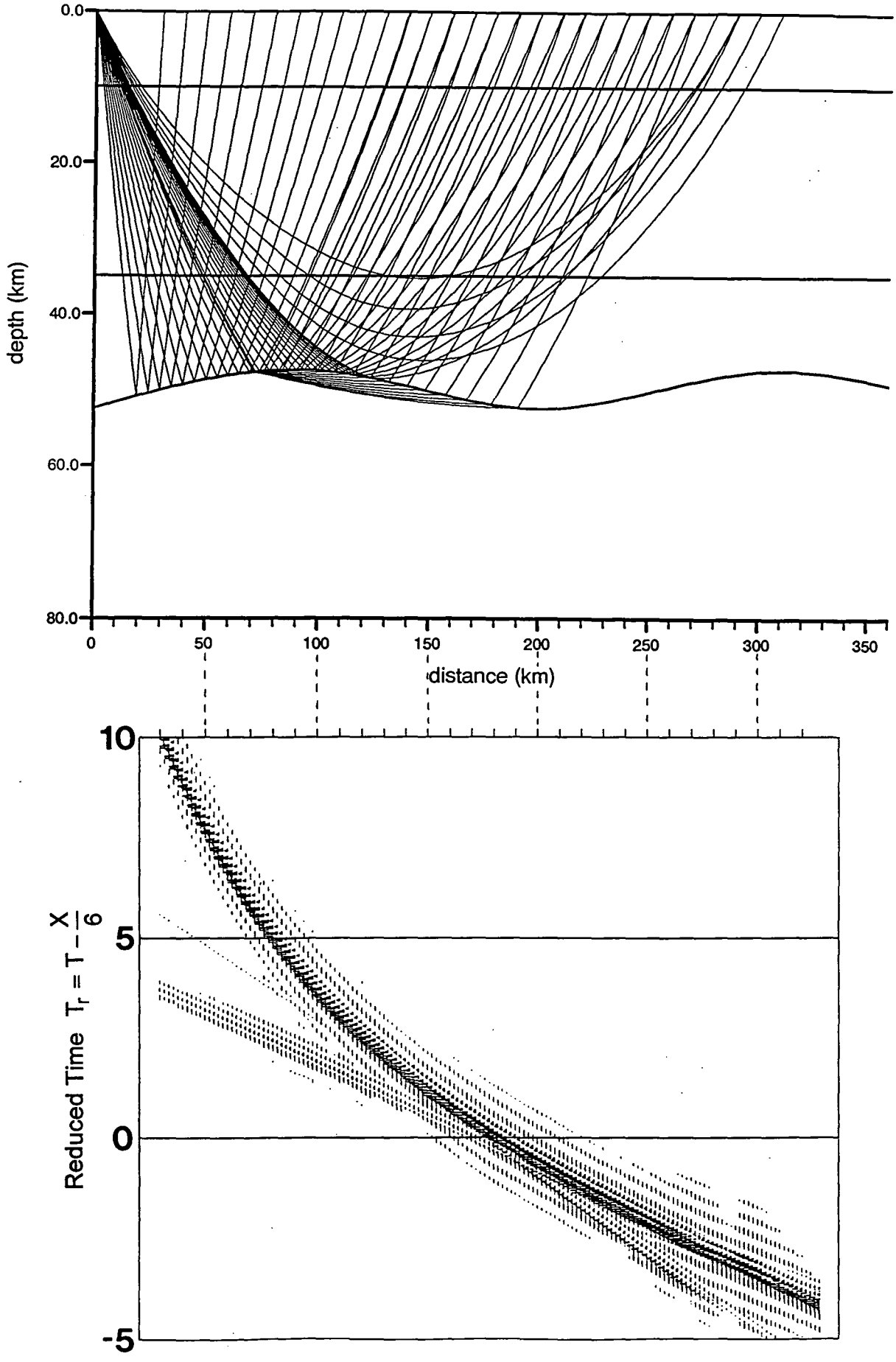


Fig. 7.4 As for Fig. 7.3 but with a long wavelength undulating Moho a) and b) show the effect of receiver location with respect to the undulation

b) Long wavelength undulations on the Moho



a) **Medium wavelength undulations on the Moho**

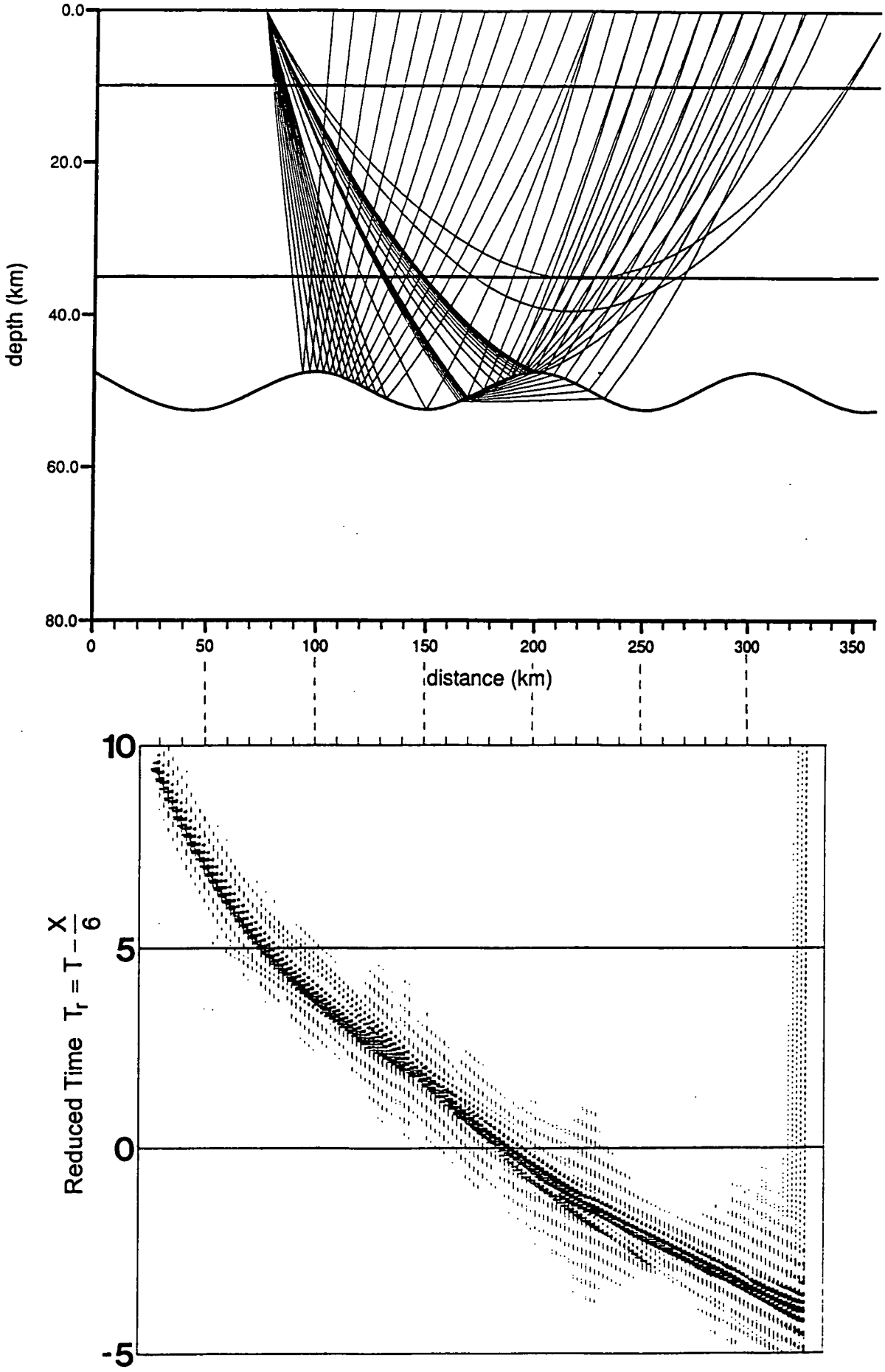
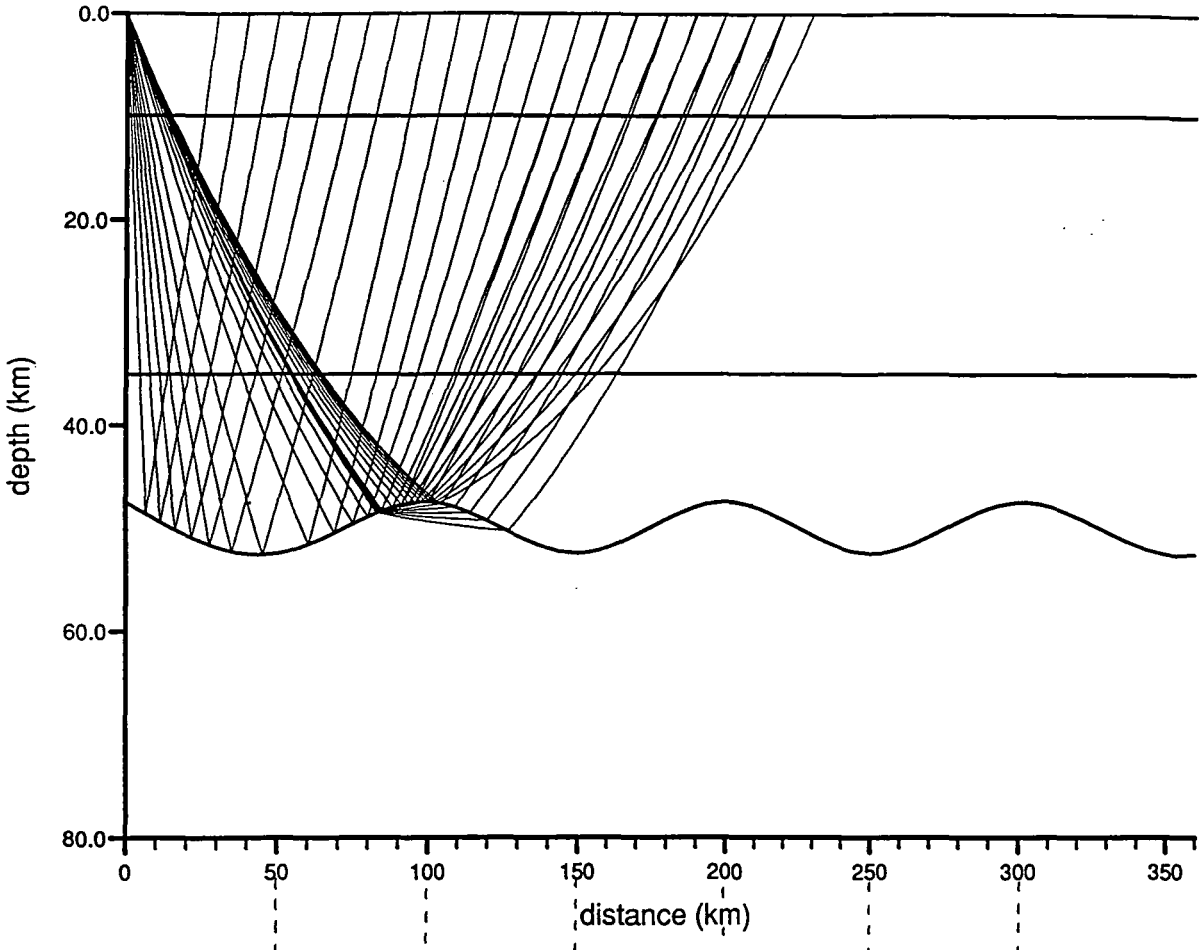
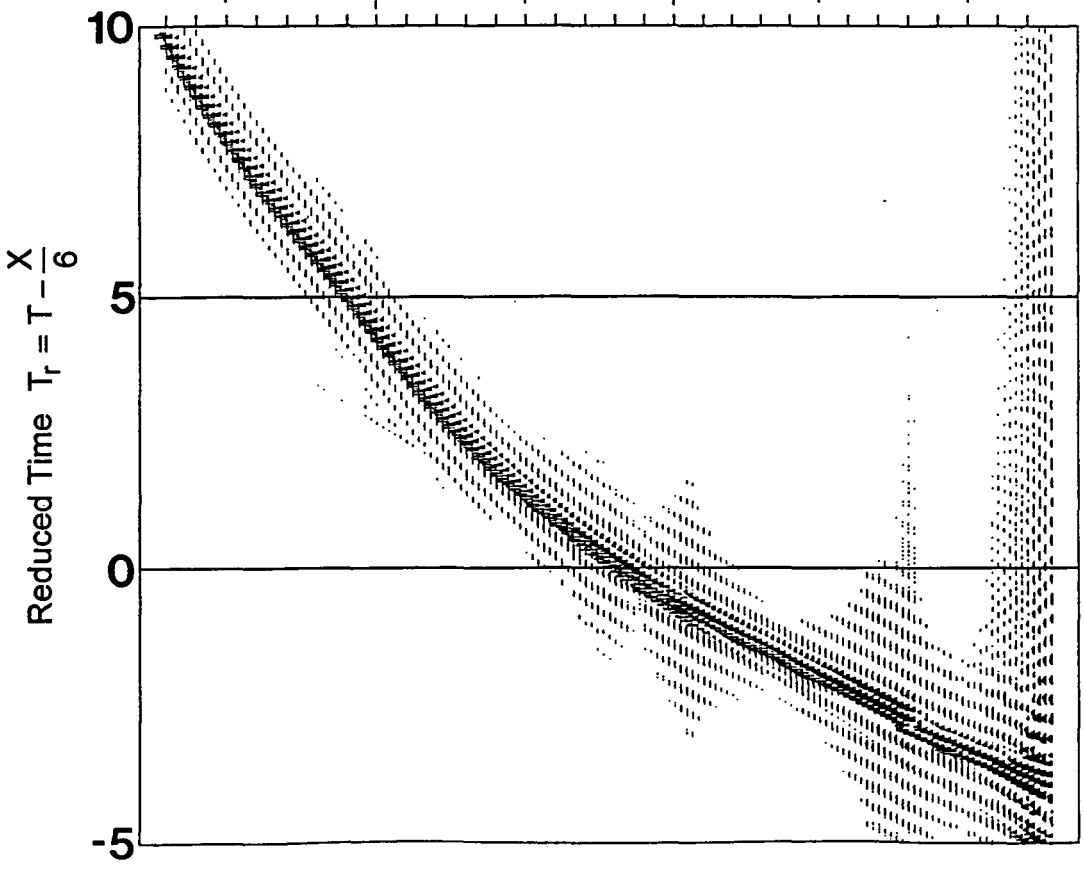


Fig. 7.5 As for Fig. 7.3 but with a medium (100 km) wavelength undulating Moho: a) and b) show the effect of receiver location with respect to the undulation

b) **Medium wavelength undulations on the Moho**





Short wavelength undulations on the Moho

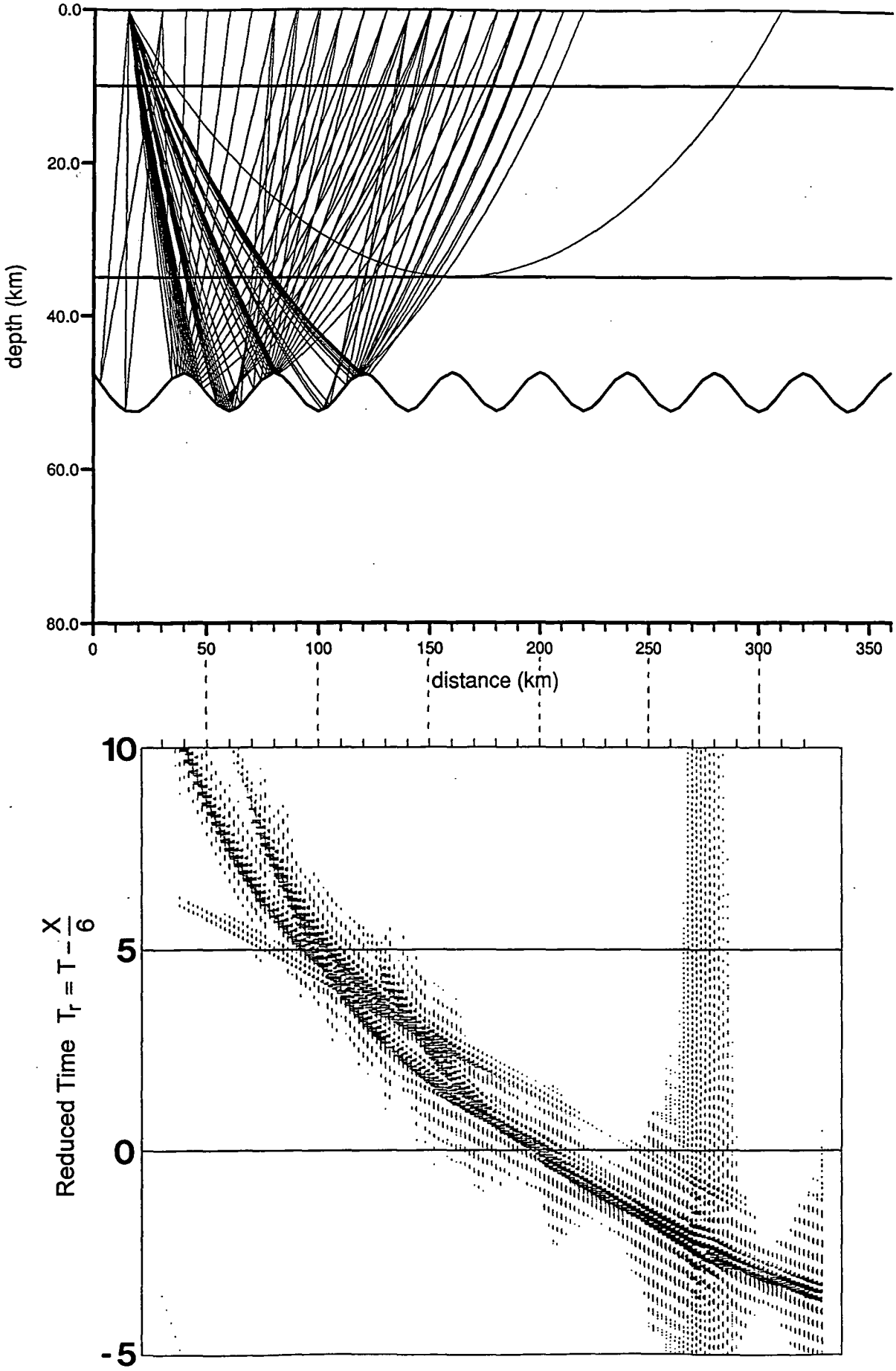


Fig. 7.6 As for Fig. 7.3 but with a short wavelength (40 km) undulating Moho

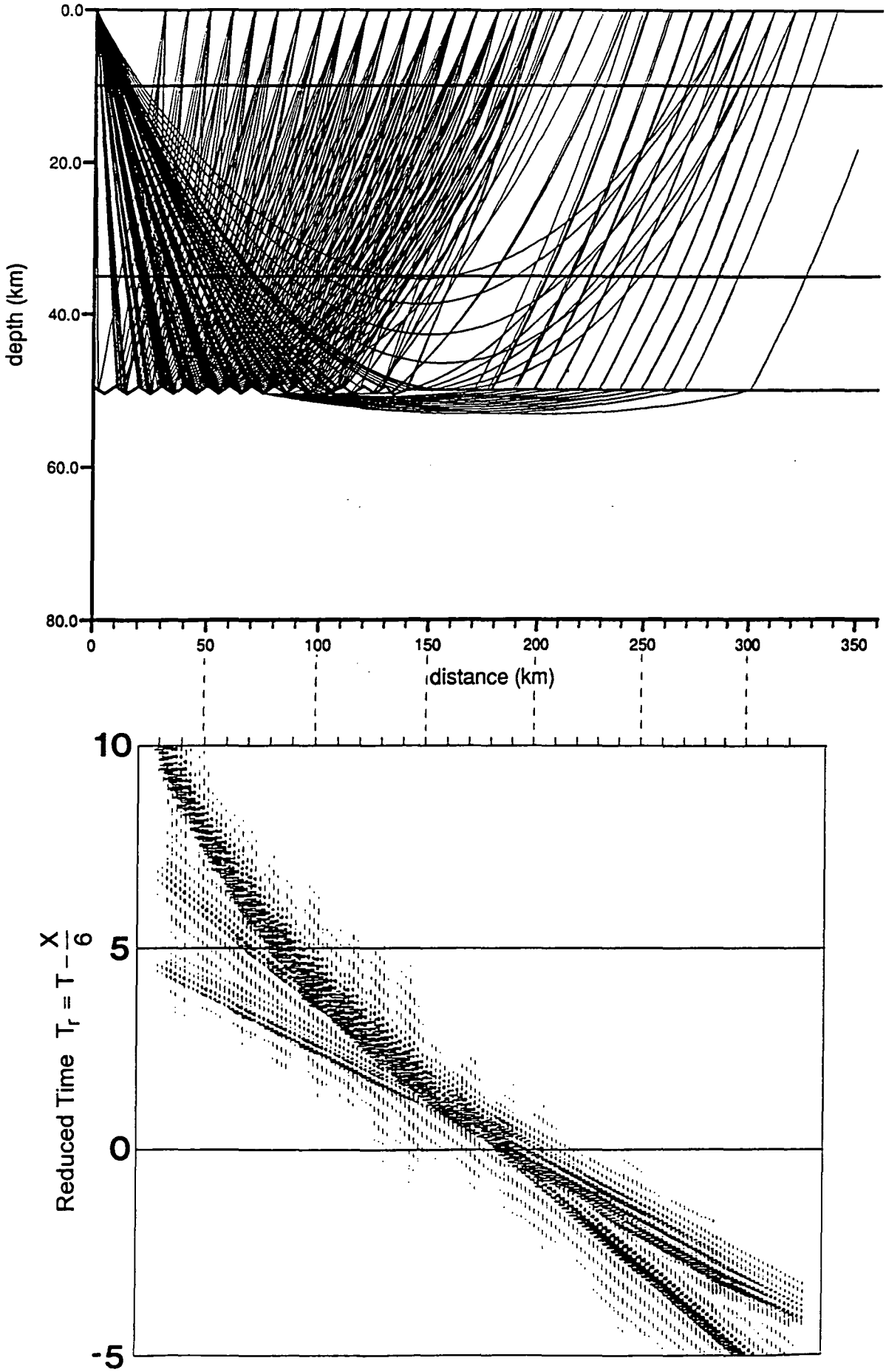


Fig. 7.7 As for Fig. 7.3 but with a very short wavelength (10 km) undulating Moho

7.3.2 Model A : 101P and F01A

The model produced for the in-line data before gravity modelling is given in Fig. 7.8. Figs. 7.9 and 7.10 show the synthetic sections generated from the model for stations 101P and F01A respectively, with Figs. 7.11 and 7.12 showing the relevant raypaths and travel-time graphs.

The model shows a fairly uniform upper and middle crust, disrupted by a low velocity body between 10 and 20 km depth in the central part of the model, and by thin reflective and diffractive bodies at 30 km depth. The central one of these thin bodies is defined only by diffractions from either end and neither are thick enough for their velocities to be determined. The lower crust is more complex, with a depression in the Moho separating two distinct lower crustal blocks.

This model was used as the basis for gravity modelling in two dimensions. The model boundaries were extended to±1000 km, and initial densities were estimated from the Nafe-Drake curve, with a reference value taken as 2970 kg m^{-3}. All density contrasts in the model are calculated with reference to this value. The crustal density contrasts were then adjusted to give the best fit to the data (Fig. 7.13). Two things are obvious from this model : (a) that the upper crustal block in the centre of the model has a high density rather than a low one as expected from its low velocity in the seismic model; (b) this same block must extend upwards, nearer to the surface in order to account for the short-width positive anomaly. Fig. 7.14 shows a model into which such an extension has been introduced. The upper crustal density contrasts have been adjusted for the best fit. It appears from this model that the high velocity upper crustal block should have greater mass towards its southern tip and

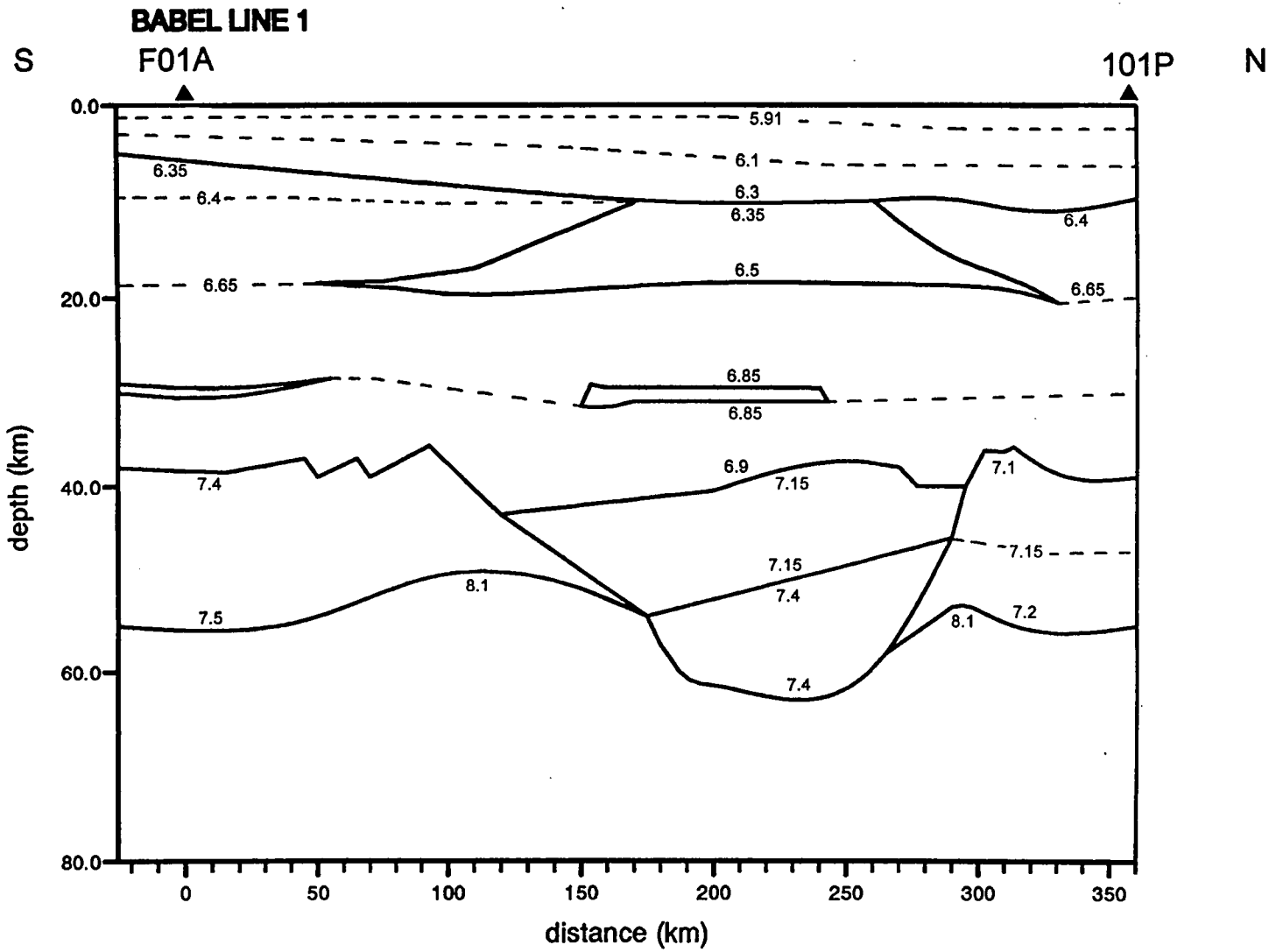


Fig. 7.8 Initial velocity model for in-line stations 101P and F01A without constraint from gravity

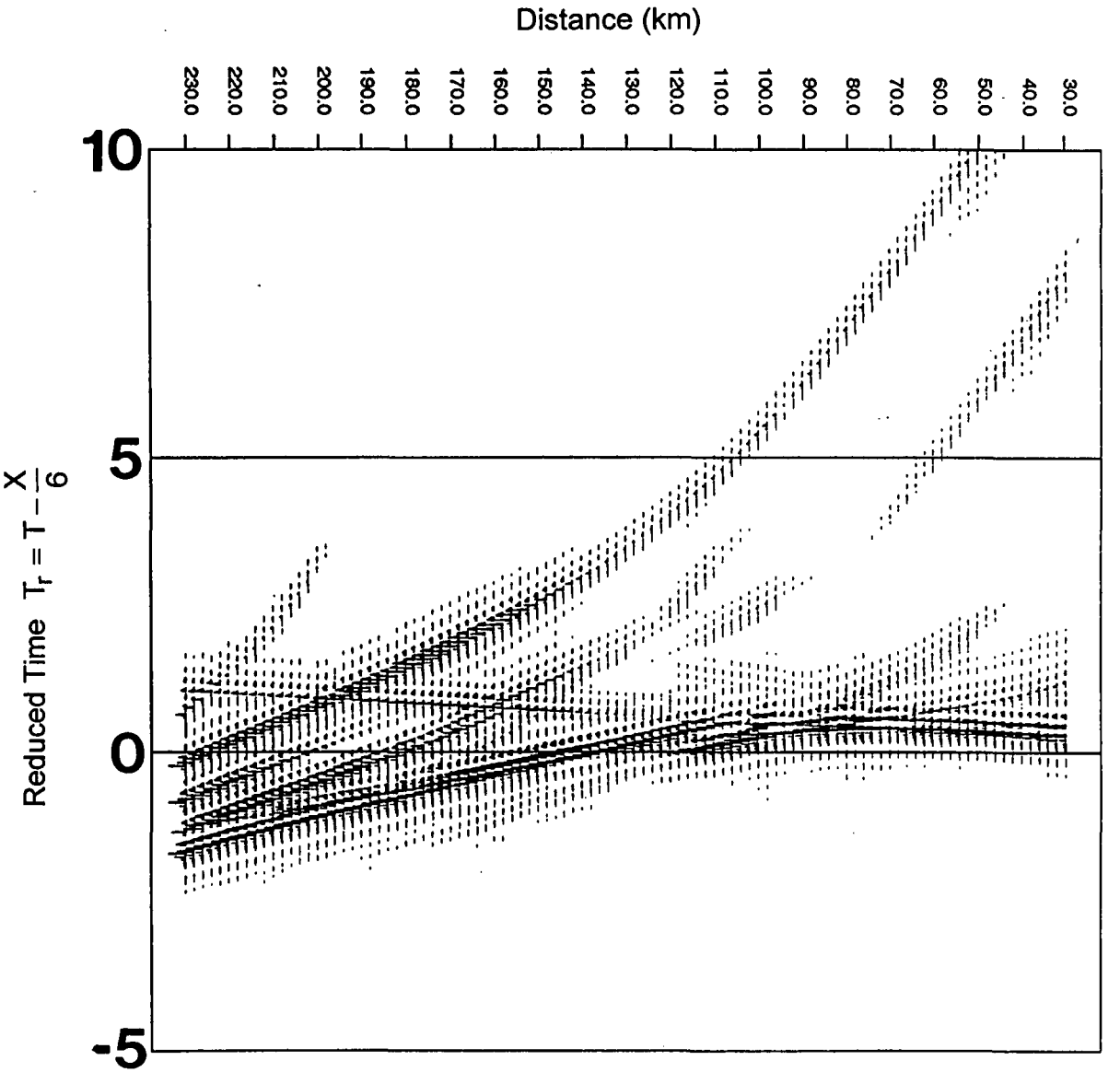


Fig. 7.9 Synthetic section generated for station 101P from initial velocity model without constraint from gravity

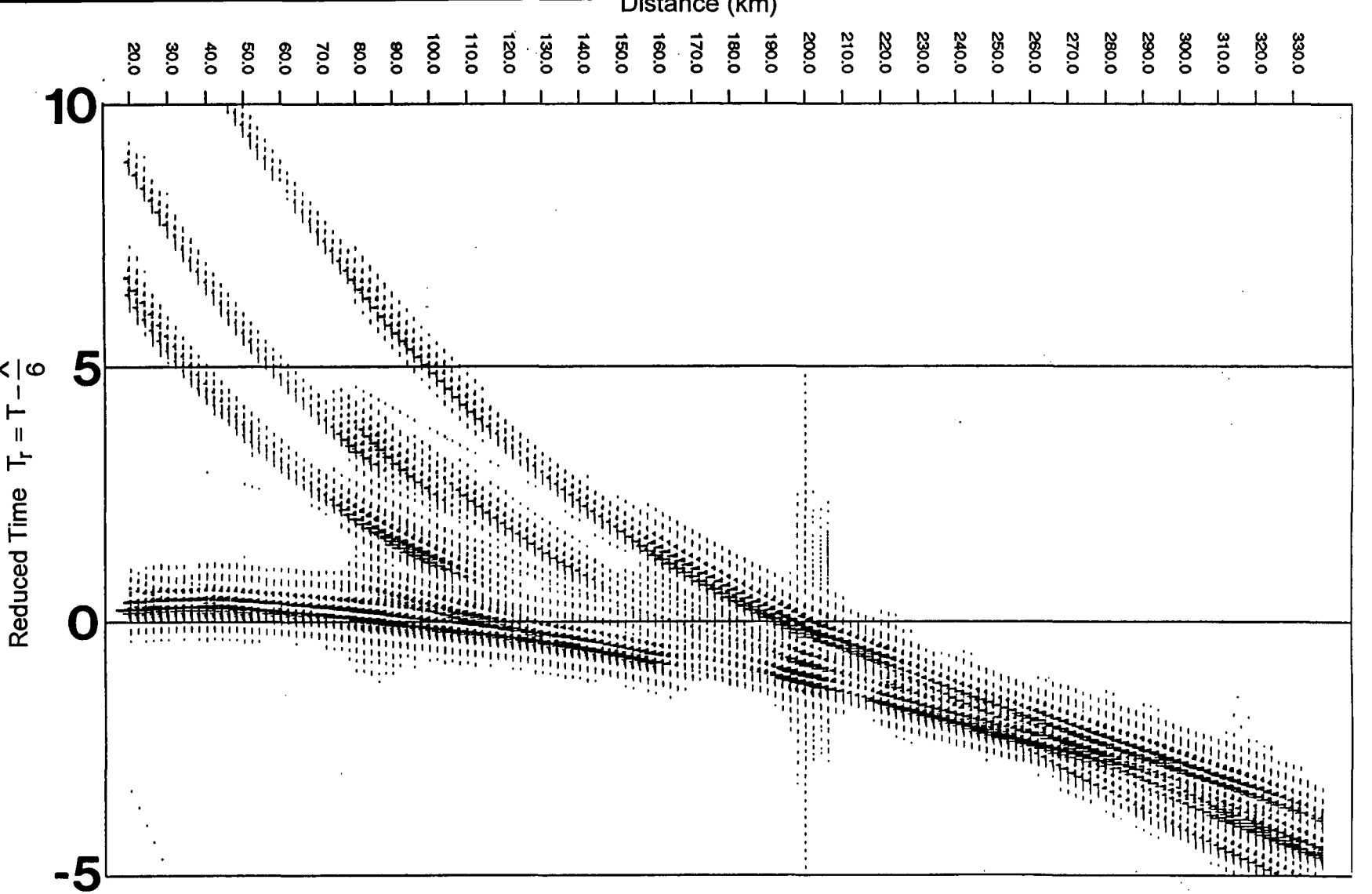


Fig. 7.10 Synthetic section generated for station F01A from initial velocity model.

without constraint from gravity

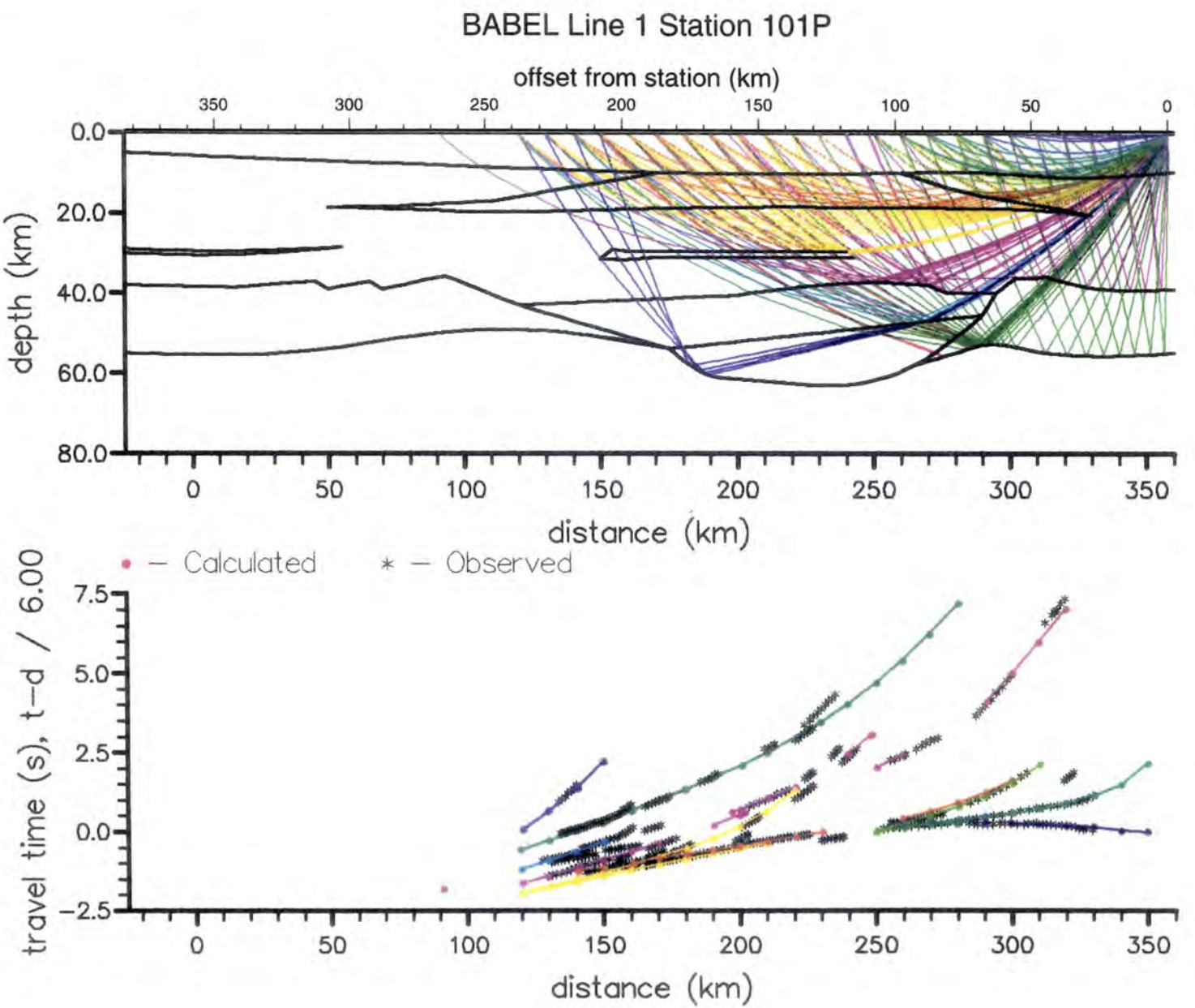


Fig. 7.11 Raypaths and travel time curves for station 101P in initial velocity model.

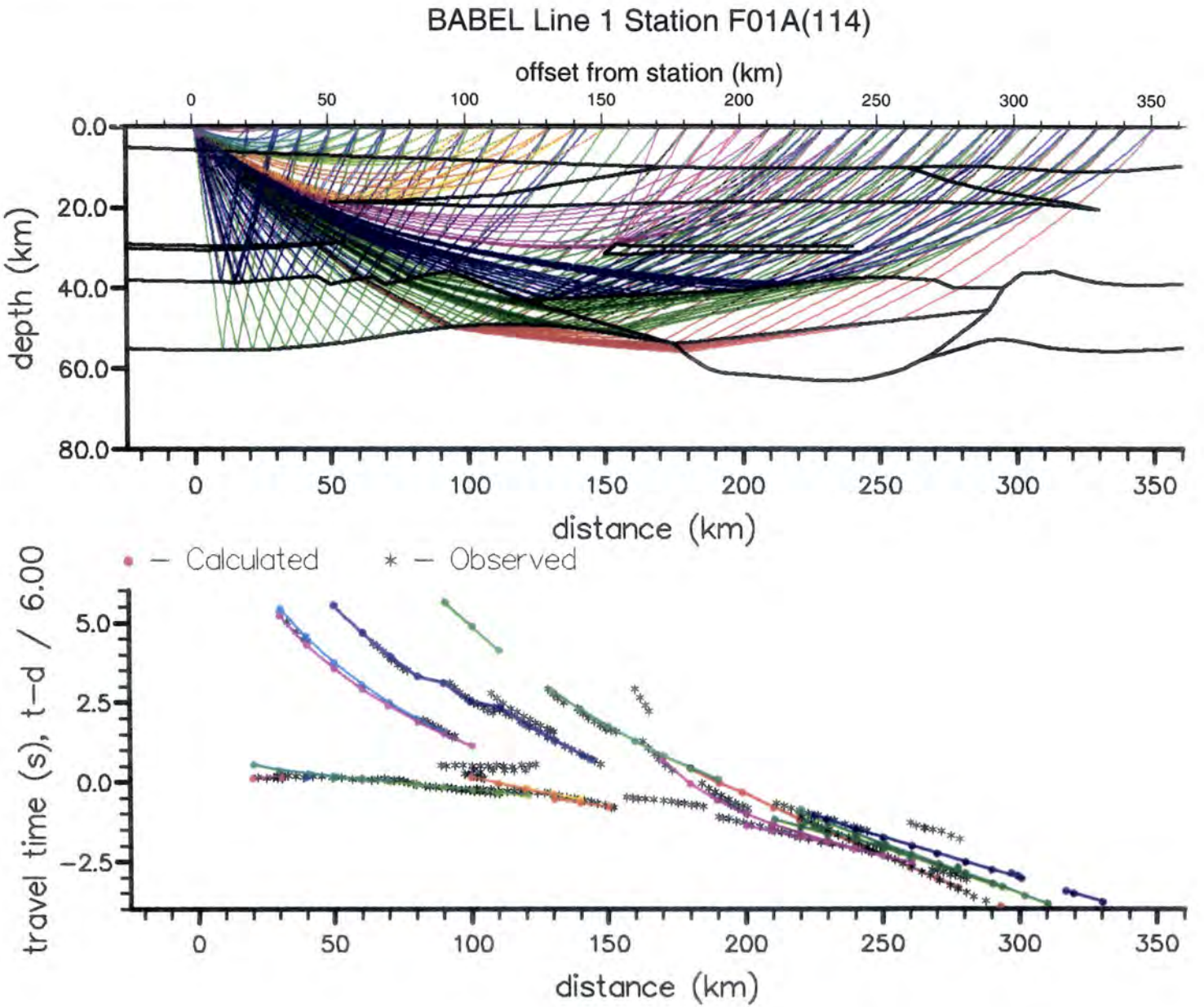


Fig. 7.12 Raypaths and travel time curves for station F01A in initial velocity model.

should not extend so far north. However, the northern upper boundary of the block is defined by the seismic model. Bearing this in mind, a low velocity block was introduced in place of the northern tip of the high velocity block with a similar upper boundary. The boundaries of these two blocks, and associated densities were altered minimally to again optimise the fit to the data (Fig. 7.15).

Referring back to the seismic data, a best fit model to both datasets was found by iteratively carrying out seismic and gravity modelling. During this process, the high density upper crustal block was reassigned a high velocity since a low velocity rock of such high density was considered unfeasible. Other boundaries and velocities below were altered to take this into account and maintain the fit to the seismic data.

Fig. 7.16 shows the final gravity model, while Fig. 7.17 shows the final seismic model. This will be referred to as Model A. The only difference between the gravity model and the seismic model which could not be resolved is the thickness of the southern tip of the high velocity/density block in the upper crust. The gravity model indicates a greater thickness here for the body than the seismic model allows. This could be accounted for if the third dimension were considered. The step in the gravity data at 300 km was not modelled since, if it was not merely an error in the processing of the data itself, its short width indicates that it is caused by a small near surface feature not imaged by the seismic data and not affecting the overall model.

The synthetic sections generated from Model A are shown in Figs. 7.18 and 7.19. They do not differ significantly from those in Figs. 7.9 and 7.10, generated from the initial seismic model, except that the diffracted arrivals are more prominent and the spurious low velocity phase at long

offsets in the section for 101P has been eliminated. The full ray diagrams and travel time curves, together with layer-by-layer breakdowns, are given in Figs. 7.20 to 7.23.

The velocity at the top of the model is 5.85 km s^{-1} and increases without discontinuity to a value of 6.3 km s^{-1}, the gradient decreasing from south to north as the depth of this first interface increases from 5 to 11 km (layer 1, Figs. 7.22a, 7.23a). Below this, the velocity increases from 6.35 at the southern end, 6.45 in the north to 6.55 km s^{-1} at a depth of between 18 and 20 km (layers 2 and 3). In the centre of the model at the same depth is a high velocity block with velocity increasing from 6.6 to 6.7 km s^{-1} (layer 4). Its southern boundary is defined by reflections of limited lateral extent (arrivals *k*, *l*, Fig. 7.23b) while its upper and northerly boundaries are obtained from gravity modelling. To the north of this is a region with a lower velocity of 6.35 km s^{-1} (layer 5). The northern sloping boundary of this region is defined by a reflected arrival (*c*) seen at station 101P (Fig. 7.22b).

At 30 km depth, the velocity has reached 6.8 km s^{-1} (layer 6). The two thin layers at the southern end at this depth (layer 7) account for laterally limited reflected phases in the data (*m*), while a high curvature reflection (*n*), interpreted as a diffraction, arises from a feature at the northern end of the more northerly of the two (Fig. 7.23c). The small body further to the north of the model at the same depth is defined only by a diffraction event (*e*, Fig. 7.22c), with no associated reflected arrival. The velocity within these bodies cannot be obtained from the data except in a qualitative way from the reflection amplitude. They have been assigned an arbitrary high velocity of 7.2 km s^{-1} but their thickness does not allow them to have a great effect on the travel times of rays passing through them.

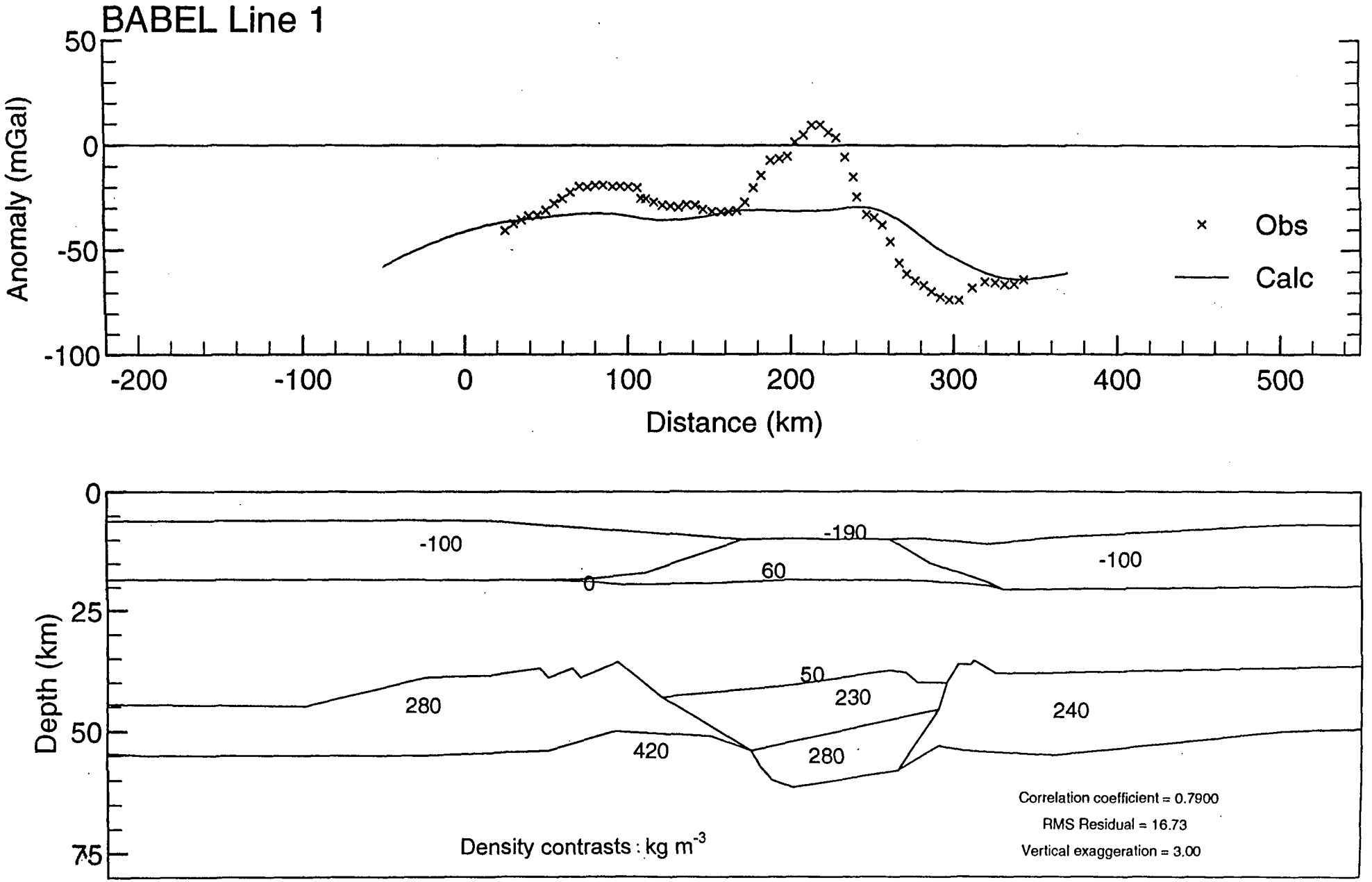


Fig. 7.13 Initial gravity model based on seismic model for Line 1.

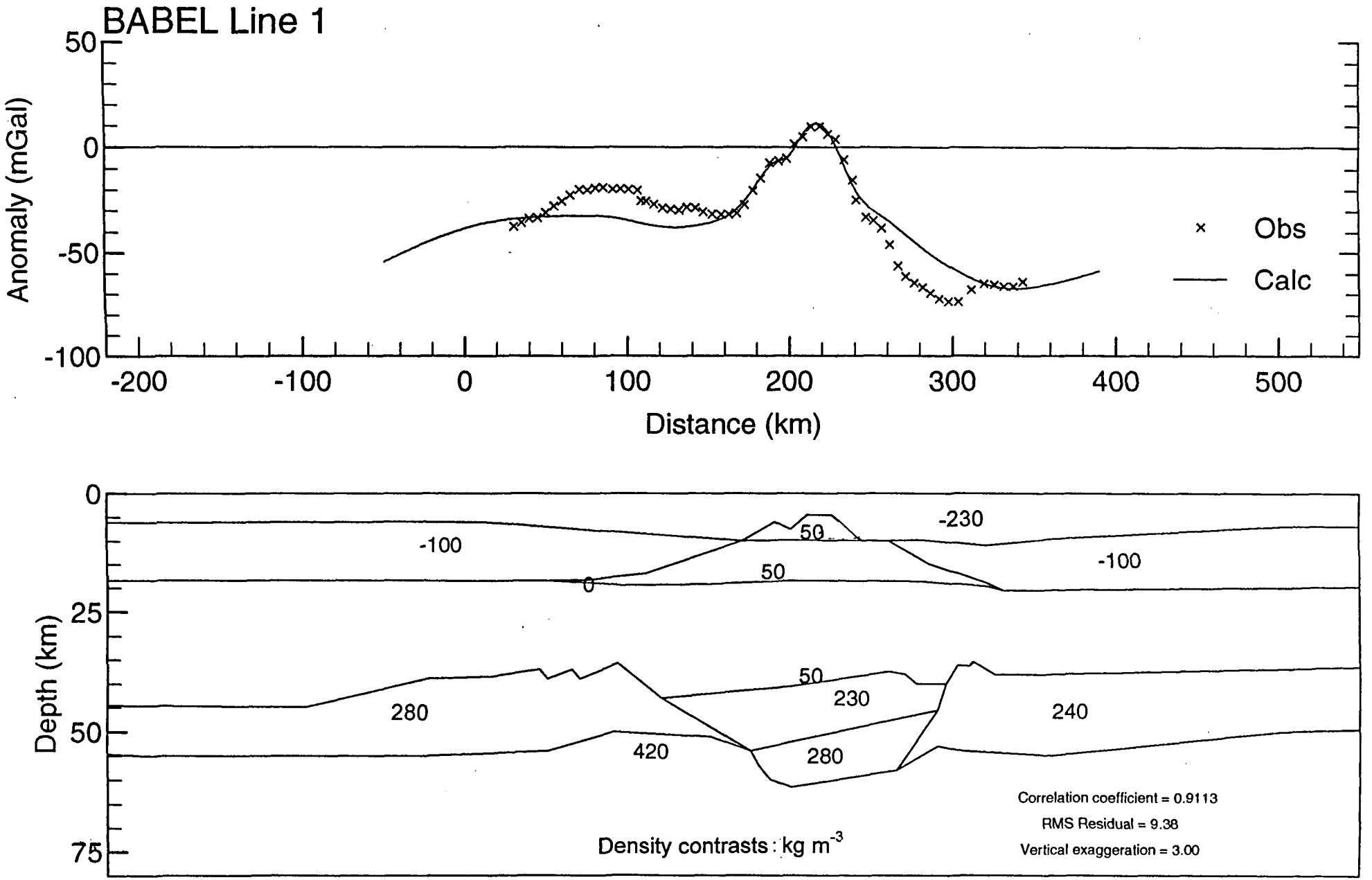


Fig. 7.14 First intermediate gravity model with additional high density upper crustal block.

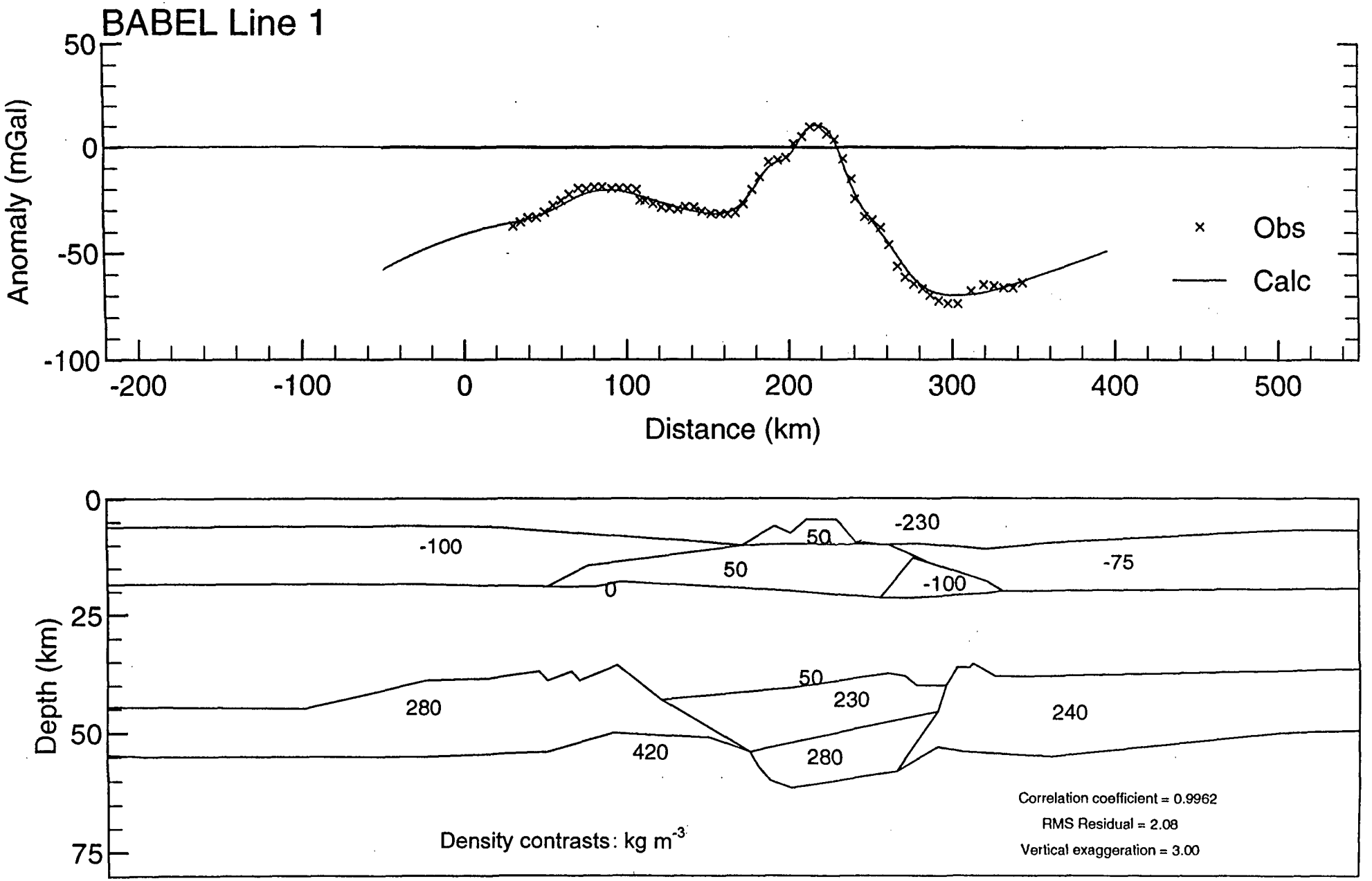


Fig. 7.15 Second intermediate gravity model with upper crustal boundaries adjusted minimally for best fit and low density upper crustal block added.

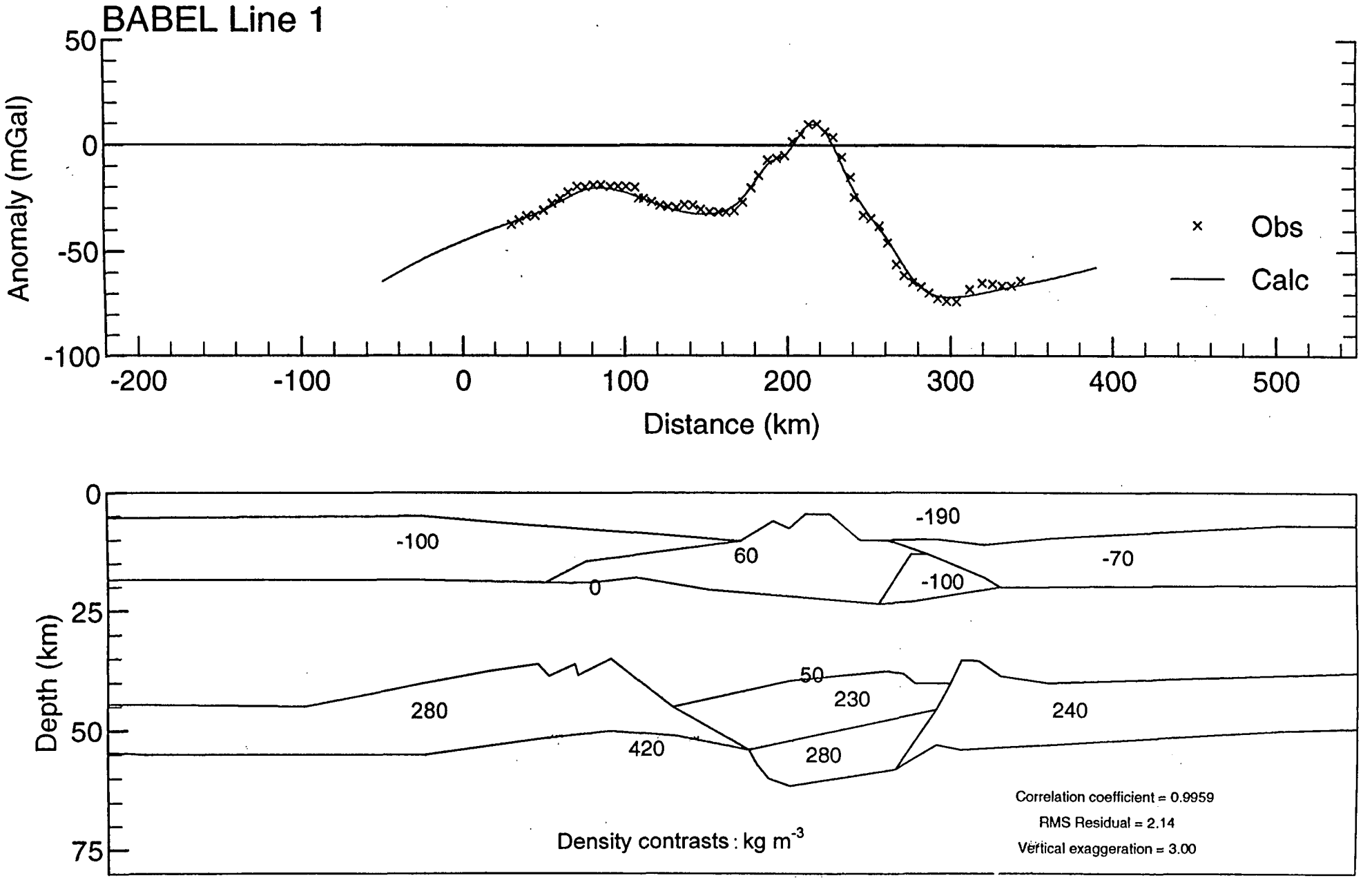


Fig. 7.16 Model A: final gravity model for Line 1.

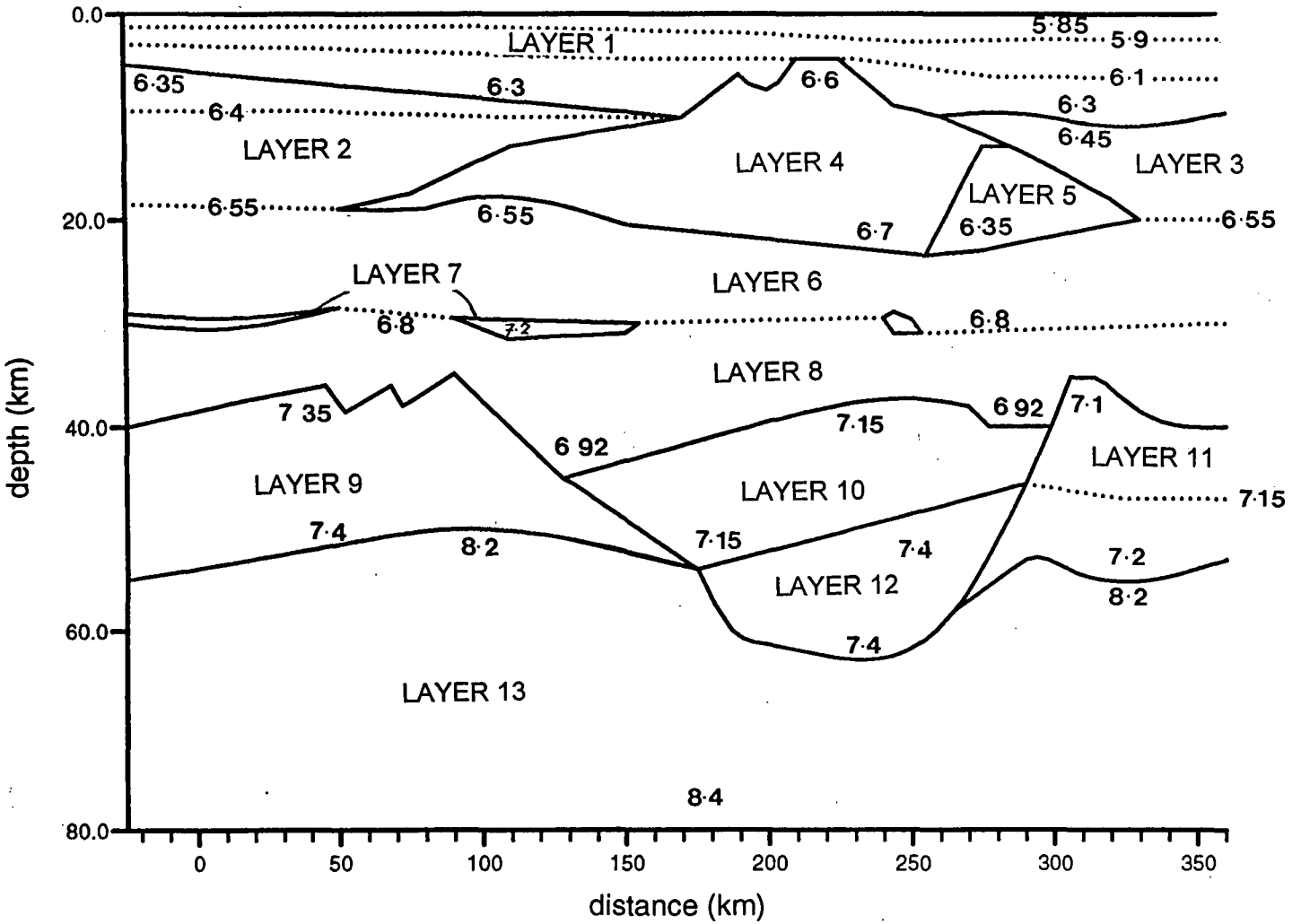


Fig. 7.17 Model A: final velocity model for in-line stations 101P and F01A.

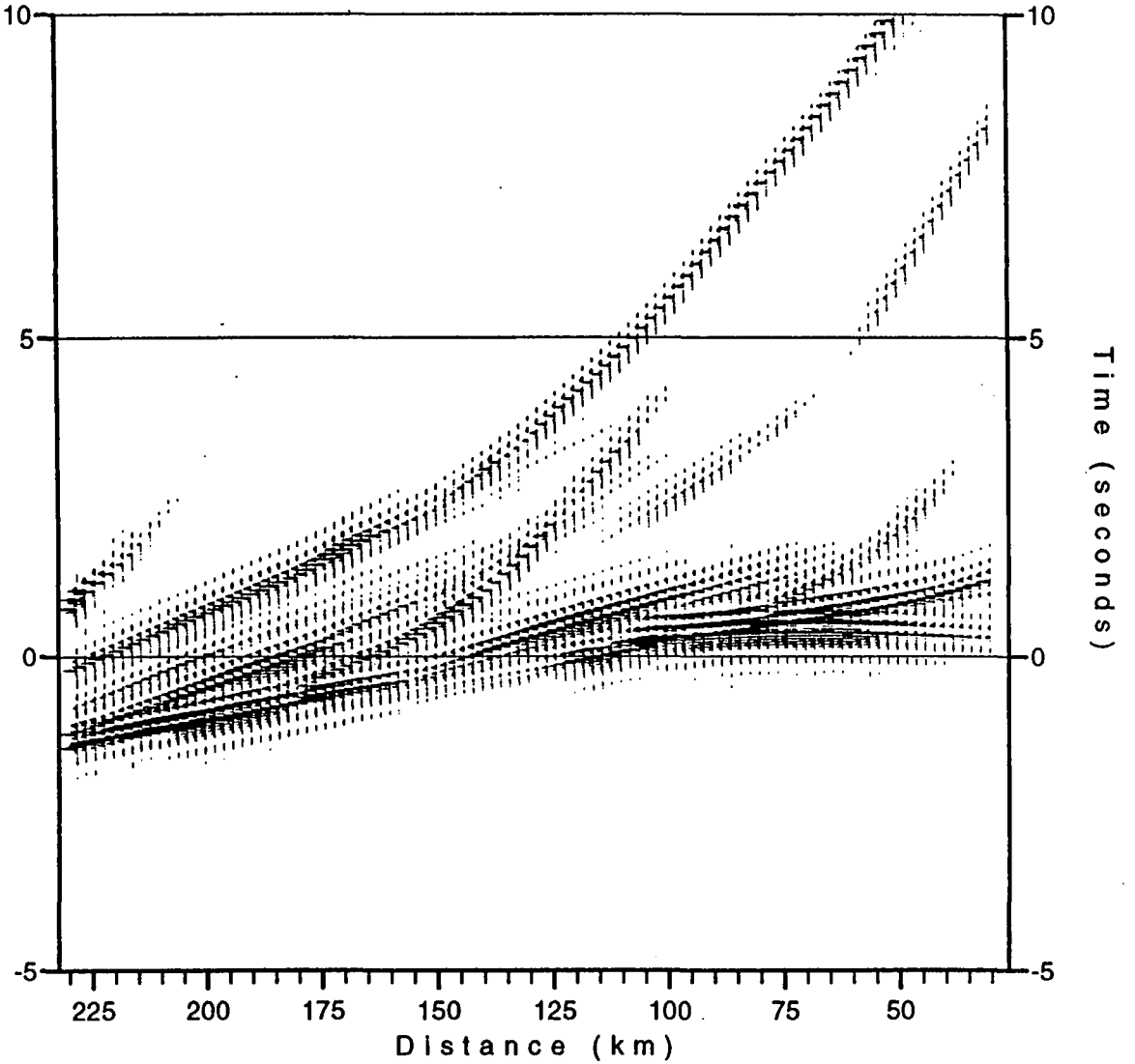


Fig. 7.18 Synthetic section generated for station 101P from Model A.

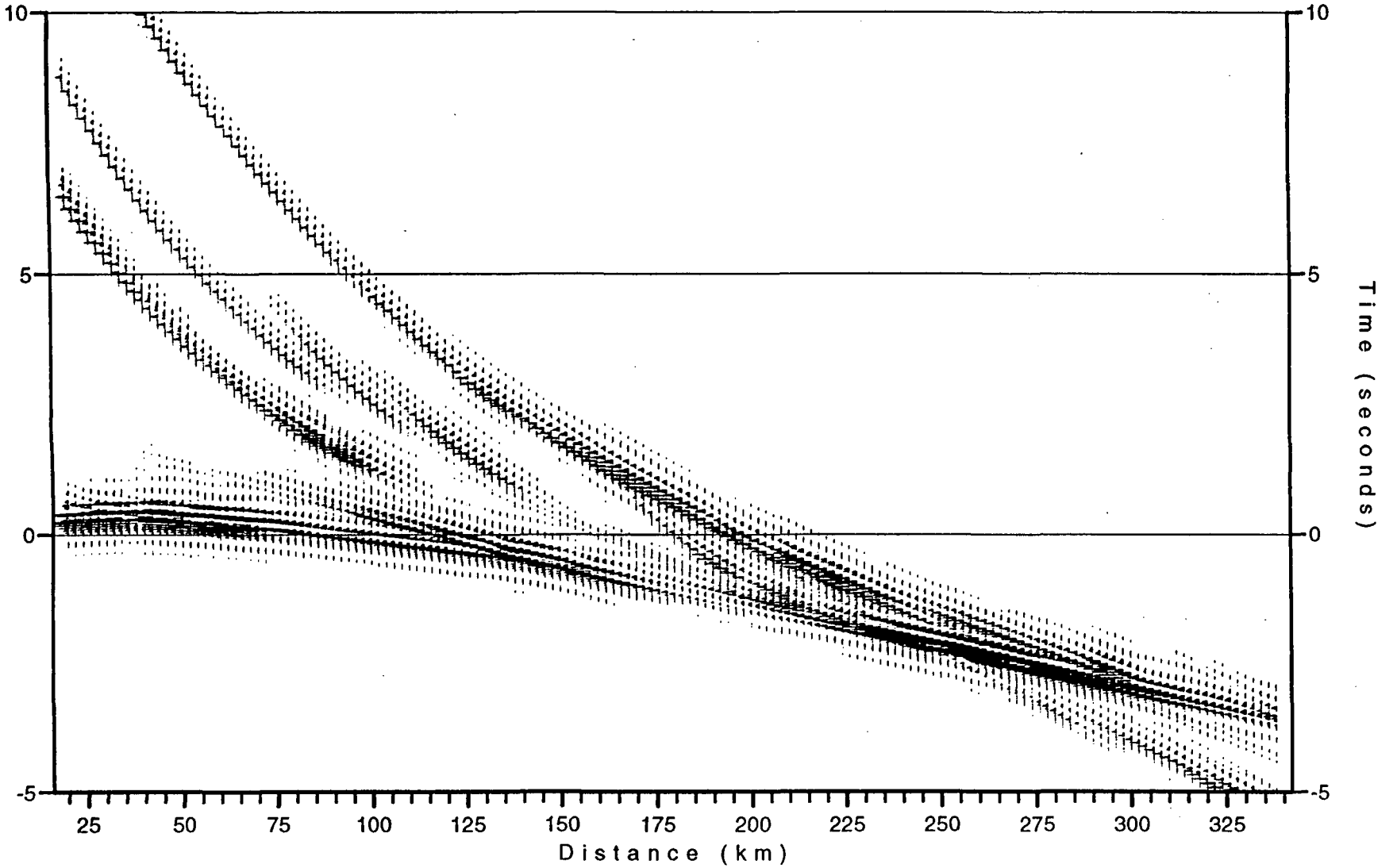


Fig. 7.19 Synthetic section generated for station F01A from Model A.

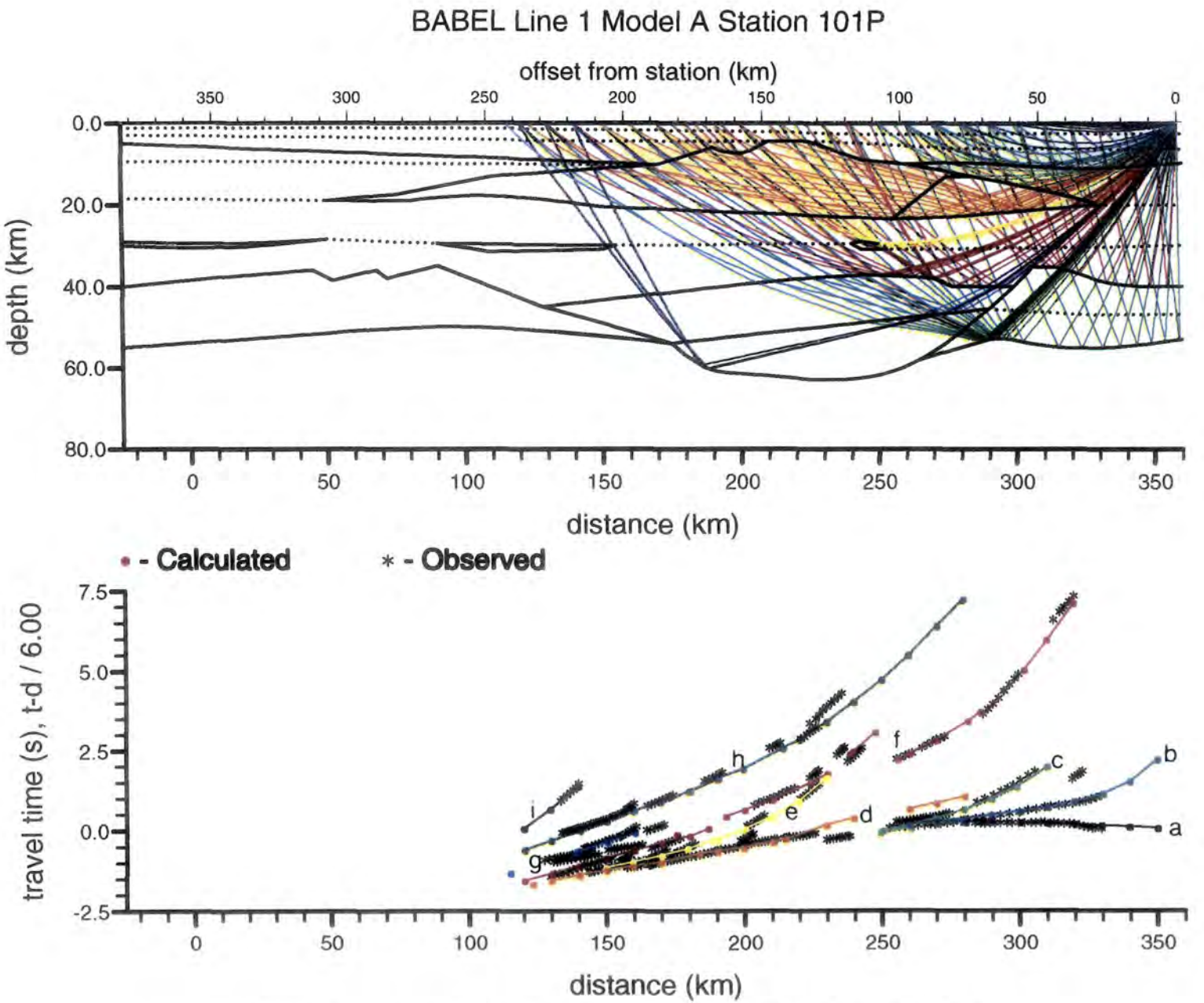


Fig. 7.20 Raypaths and travel time curves for station 101P, Model A.

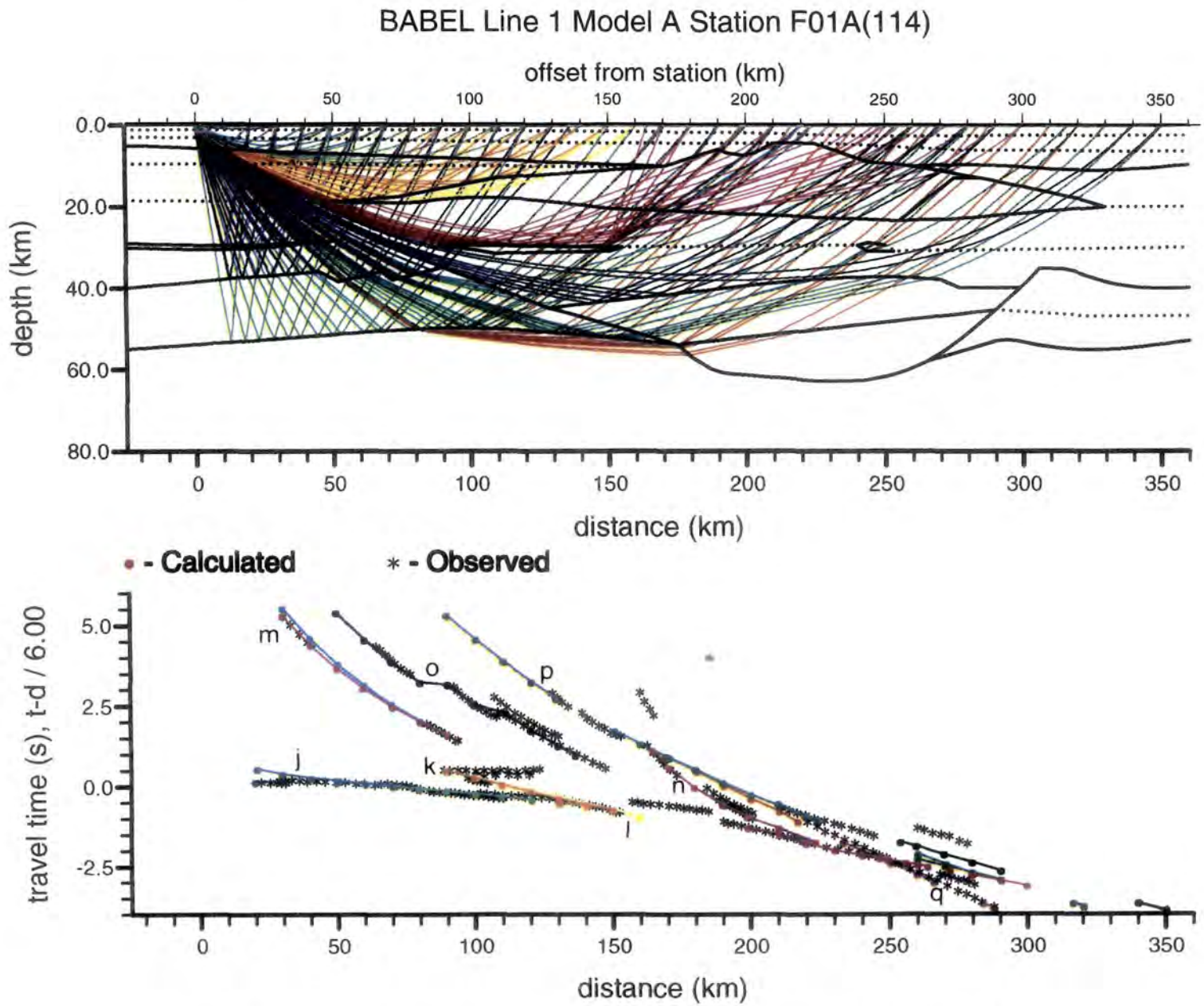


Fig. 7.21 Raypaths and travel time curves for station F01A, Model A.

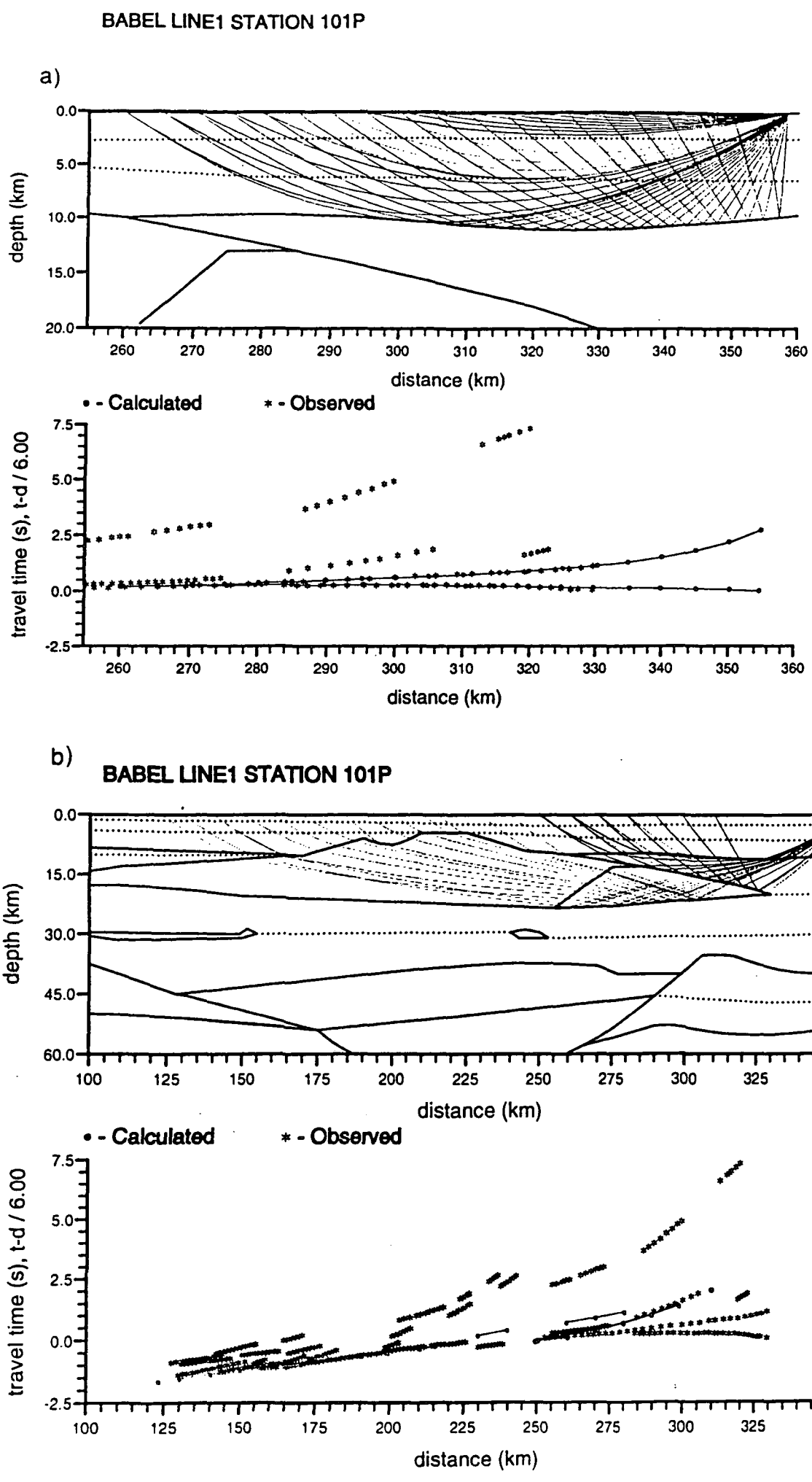


Fig. 7.22 Breakdown of raypaths and travel time curves for station 101P

a) Layer 1

b) Layers 3 and 5

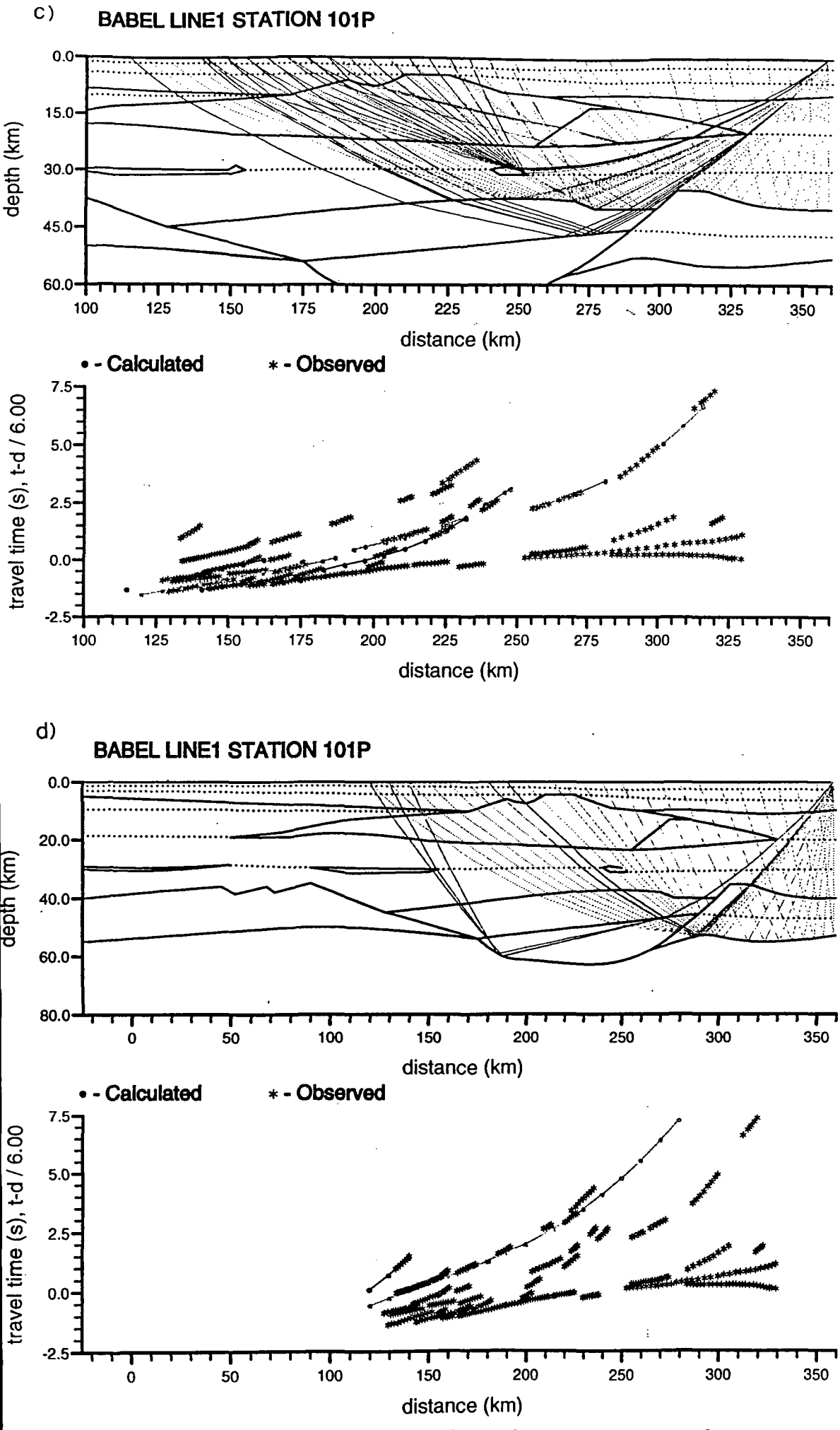
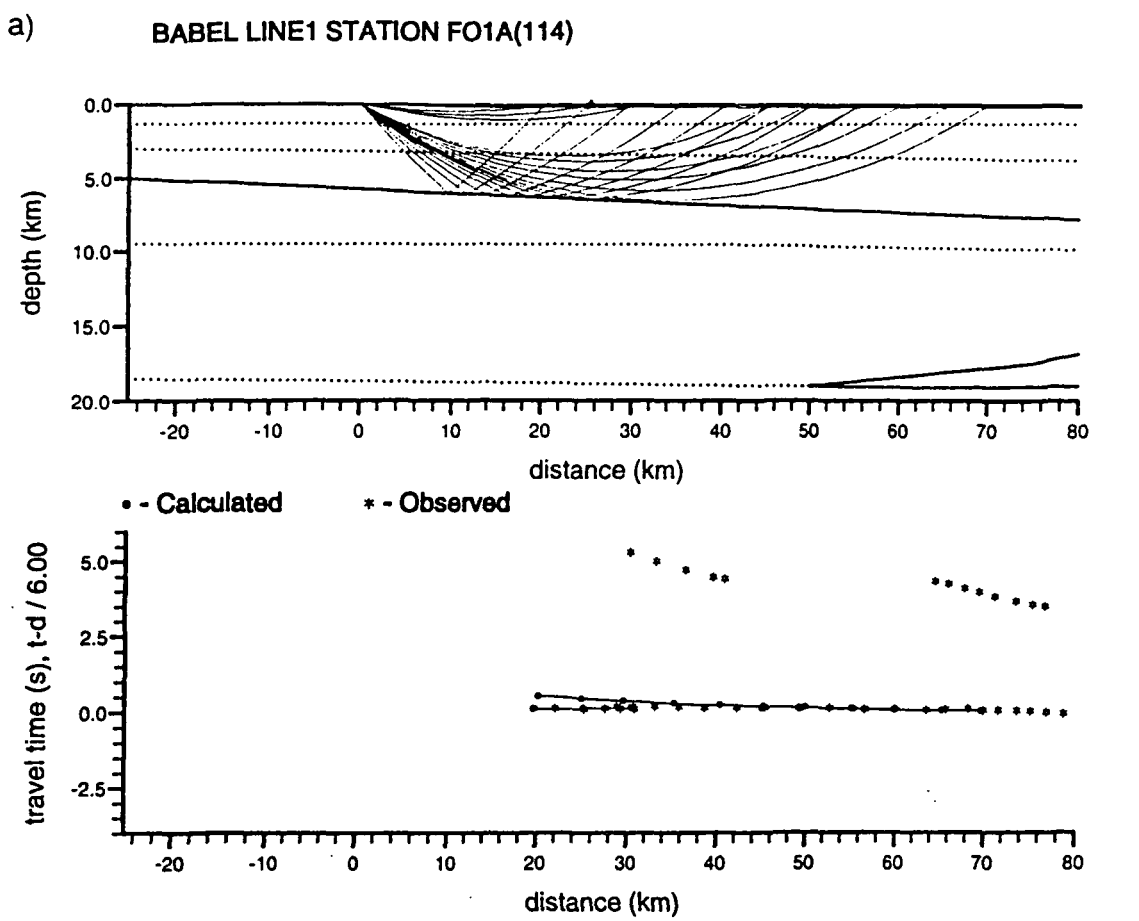


Fig. 7.22 (cont.) Breakdown of raypaths and travel time curves for station 101P

c) Layers 6, 8 and 10

d) Layers 11 and 12



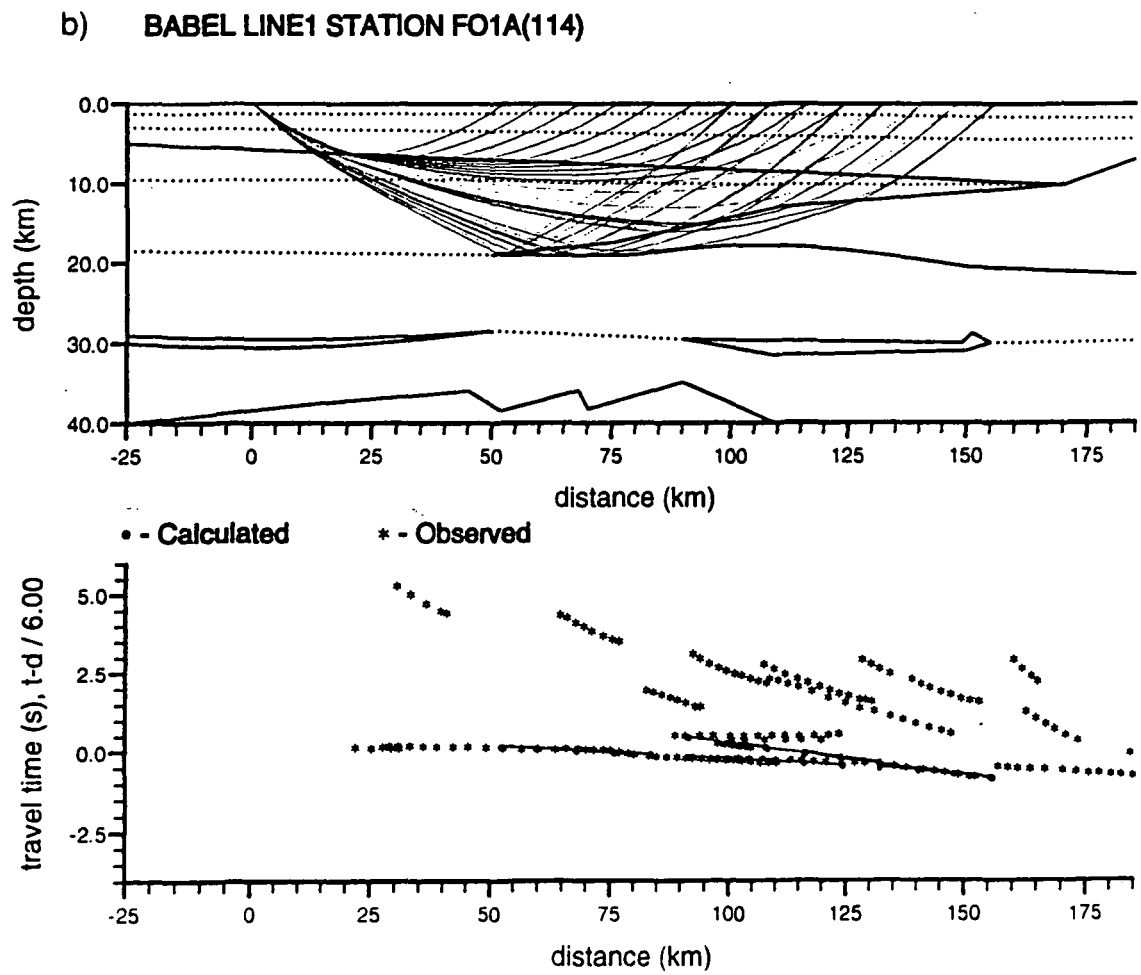
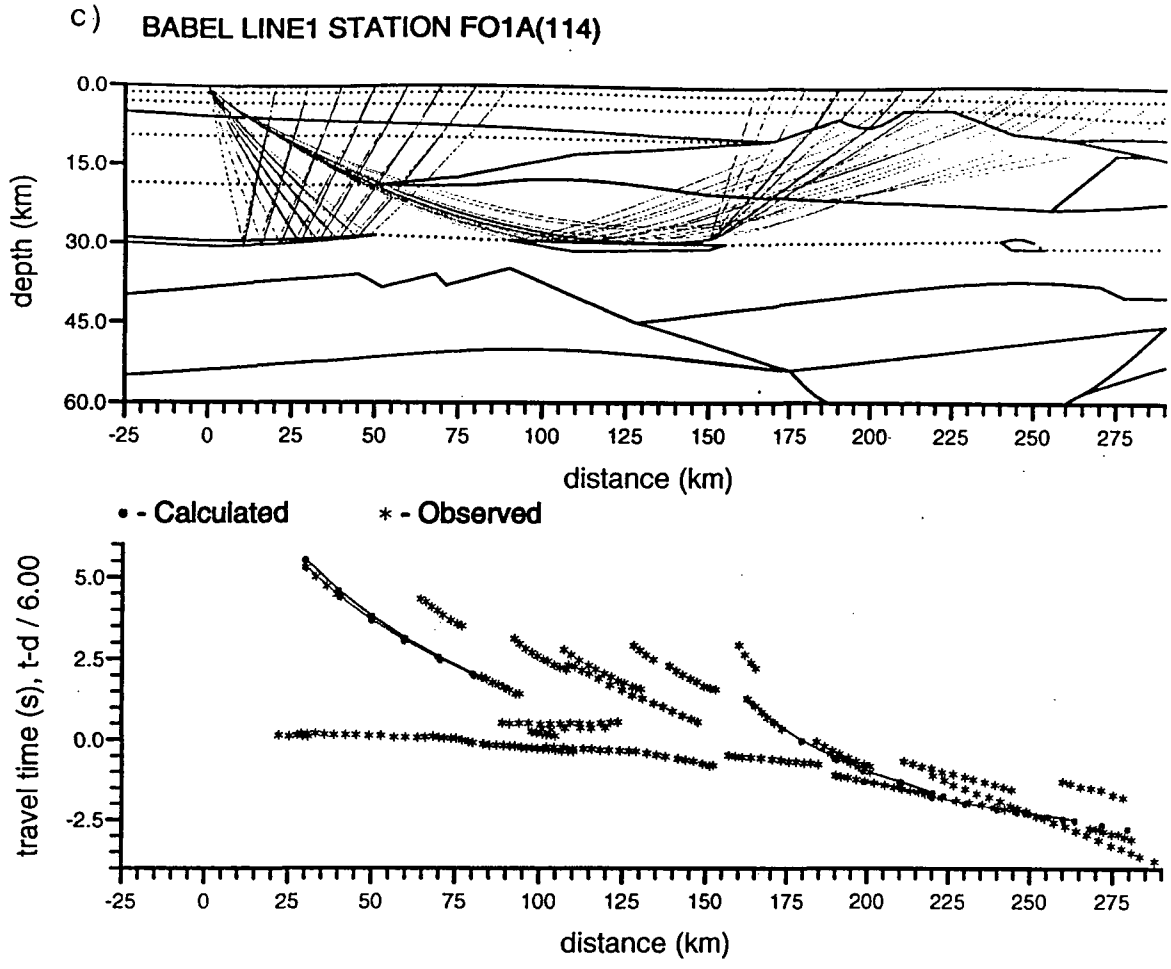


Fig. 7.23 Breakdown of raypaths and travel time curves for station F01A.

a) Layer 1

b) Layers 2 and 4



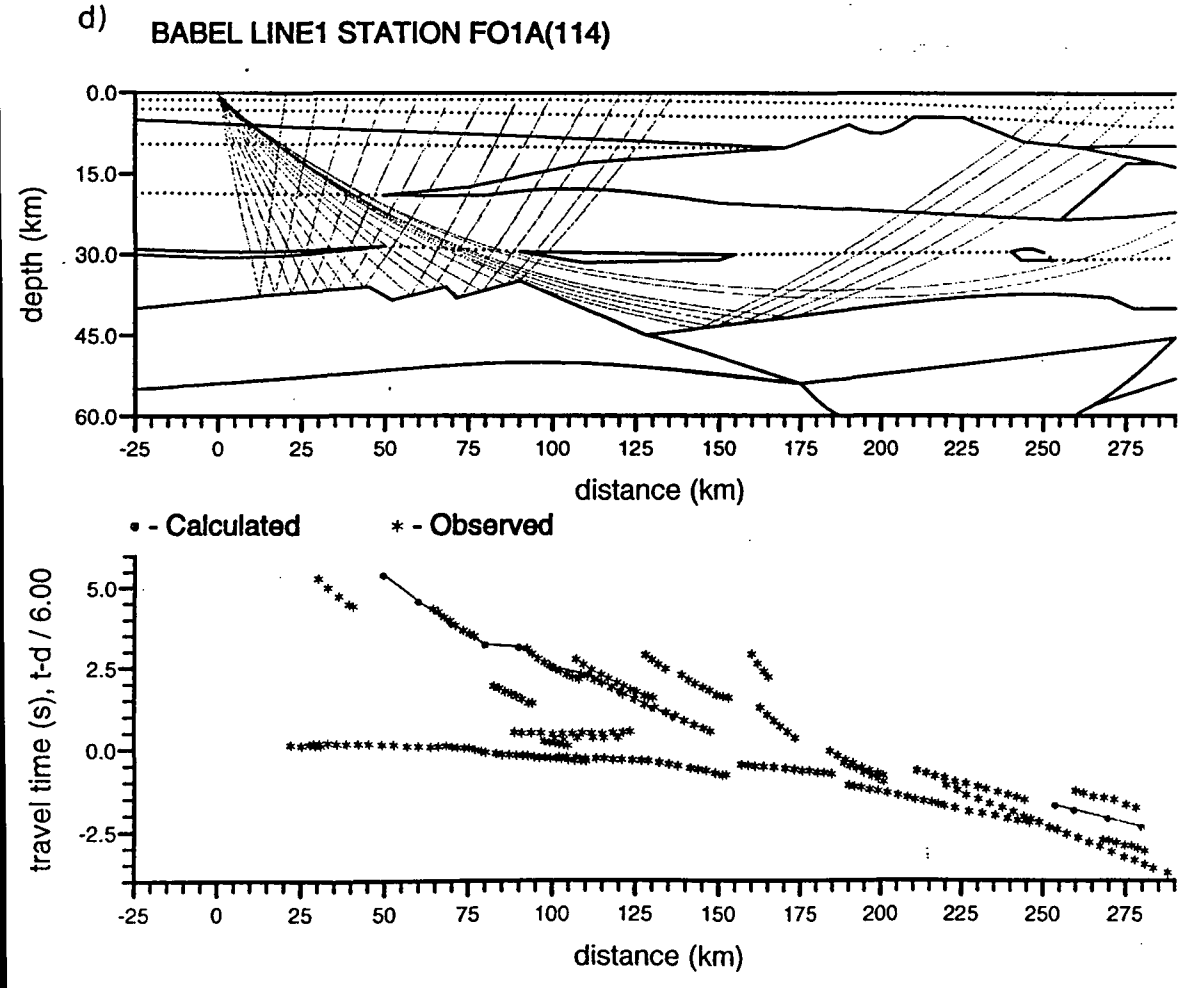


Fig. 7.23(cont.) Breakdown of raypaths and travel time curves for station F01A.

c) Layer 6

d) Layer 8

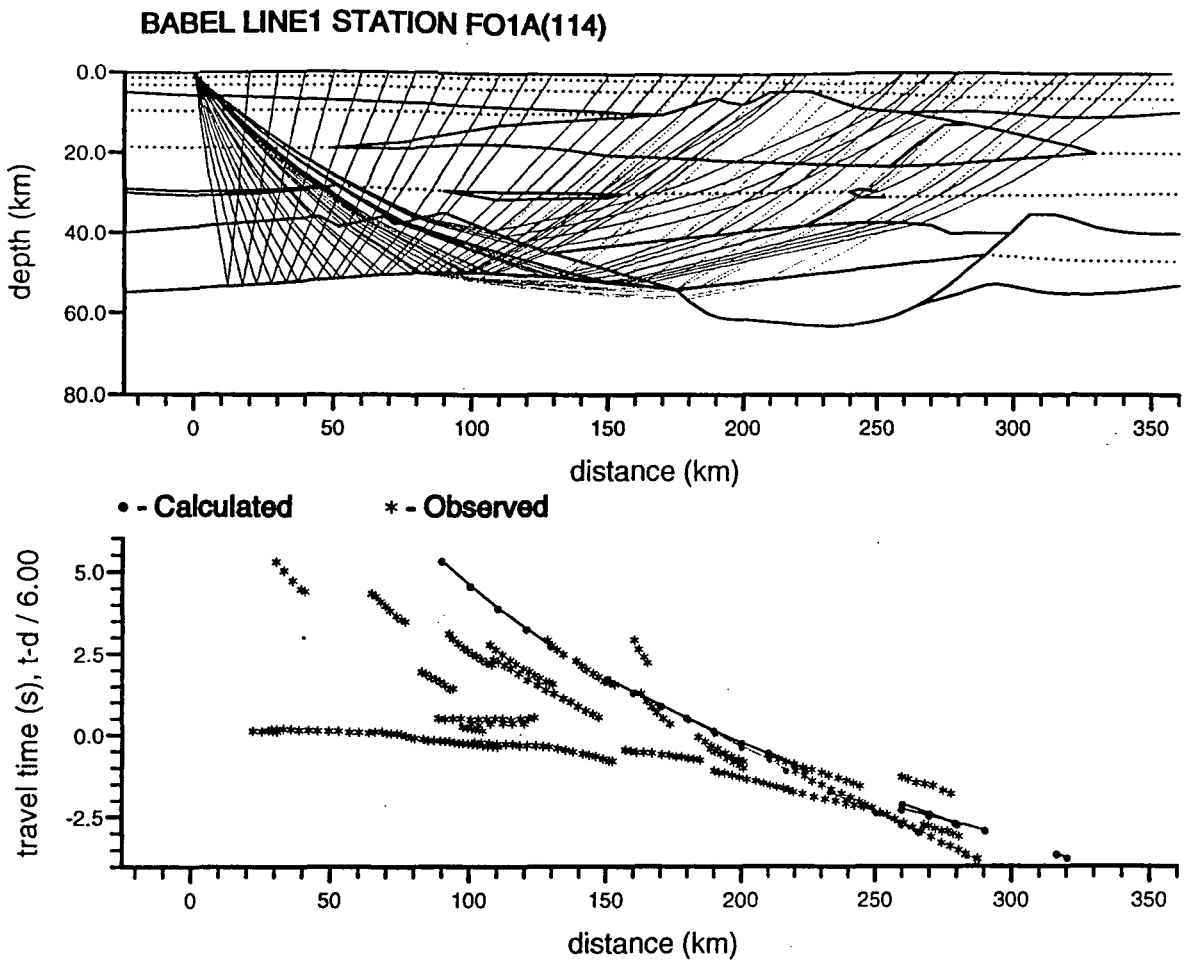


Fig. 7.23(cont.) Breakdown of raypaths and travel time curves for station F01A.

e) Layers 9 and 13

These bodies also mark the lower limit of crustal diving rays observed in the data.

The lower crust is the most complicated part of the model. At station F01A two arrivals are seen which are interpreted as lower crustal reflections (*o* and *p*). The earlier of these (*o*) appears to be segmented, made up of shorter arrivals of slightly greater apparent velocity than the general trend of the energy. This has been modelled using a jagged interface (boundary between layers 8 and 9, Fig. 7.23d). The high amplitude of this arrival indicates a large velocity contrast across the boundary. However, no diving ray is associated with layer 9 below this boundary. The second of the two reflected phases (*p*) seems to arise from the base of the crust as it ties in with the subcrustal diving ray (*q*), although its amplitude is lower at nearer offsets than the first (Fig. 7.23e).

From the northern end, again two major lower crustal arrivals have been modelled (*f* and *h*). The first of these (*f*) is the lower in amplitude and is in fact only seen as short coherent segments in the data. To model this it was necessary to have high relief on the reflecting boundary (boundary between layers 8 and 10,11, Fig. 7.22c). It is not certain that all the arrival segments arise from a continuous boundary. The second arrival (*h*) is a reflection from a deeper boundary. This has been interpreted as the Moho although no subcrustal diving rays are seen from station 101P to verify this due to the loss of data beyond 230 km offset (Fig. 7.22d). This interpretation is not unreasonable due to the arrival's high amplitude and lower crustal origins. The Moho reflection is often regarded as the highest amplitude long offset arrival on the record section though the strict definition of the Moho is the point where the velocity exceeds 8 km s^{-1}.

The latter arrival (*h*) is preceded by a shorter parallel arrival of similar amplitude which only appears beyond an offset of 180 km (*g*). It is therefore modelled by an interface at depth ~48 km (between layers 10 and 12) which ends at 290 km in the model (Fig. 7.22c). It is interesting to note that the Moho reflection arises from only a short length of the actual boundary between 270 and 300 km. A high apparent velocity arrival at the southern end of the 101P record section (*i*) is modelled as a reflection from a steeply dipping part of the Moho representing the far wall of a trench (Fig. 7.22d).

The subcrustal diving ray (arrival *q*) recorded at station F01A indicates that the velocity below the Moho is 8.2 km s^{-1} (Fig. 7.23e).

The overall impression given by the model is of a three layer crust cut by a wide disrupted zone. The middle and upper crust are similar at the northern and southern extremes of the model while the seismic velocity of the lower crust appears to be higher at the southern end than at the northern end. The central segment is a combination of the two.

7.3.3 Model B : Station 1A

When the two off-line stations (1A and 5A) were added to Model A in order to constrain it further, it became clear that they would have to be modelled separately, the upper- and mid-crustal arrivals being quite different.

The nearest offset of data for station 1A is 100 km so no structure is determined for most of the upper crust. The velocities have been chosen so that the apparent velocity and travel time of the first observed arrival is correct. Figs. 7.24 and 7.25 show the velocity model and synthetic section

for station 1A respectively while Figs. 7.26 and 7.27 shows the raypaths and travel time graphs. The velocity increases steeply from 5.95 km s^{-1} to 6.2 km s^{-1} at a depth of 3 km and then with a shallower gradient to 6.4 km s^{-1} at 10 km and 6.65 km s^{-1} at 18 km (layer 1, arrivals *a* and *b*, Fig. 7.27a). This last interval is similar in velocity and gradient to the northern and southern extremes of Model A described above.

Below 18 km there is a lower velocity zone, the velocity dropping to 6.45 km s^{-1} across the boundary and then rising to 6.5 km s^{-1} at 24 km and 6.75 km s^{-1} at an interface which increases in depth from 30 to 36 km between 200 and 300 km in the model (layer 2). The decrease in velocity at 18 km depth is required to explain the step out in the first arrival at 170 km offset (phases *b* and *c*, Fig. 7.27b).

The next interface also dips gently to the north from 43 to 47 km depth, the velocity above reaching 7.0 km s^{-1}. At 190 km along this boundary there is an irregularity, represented by a hump, from which a diffraction-like event occurs (arrival *e*). The boundary to the south of this feature is not defined by reflected arrivals in the data (Fig. 7.27c). A similar feature occurs on the Moho at 55 km depth, 175 km along the model (arrival *g*, Fig. 7.27d). The velocity above the Moho increases from 7.4 to 7.6 km s^{-1}, similar to the southern end of Model A. A third diffracting event (*i*) occurs in the upper mantle at a depth of around 65 km, 135 km along the model (Fig. 7.27e). No reflecting boundary is seen to the north of this so it has been modelled as the northern end of a thin low velocity body (layer 6). Laminations of the upper mantle have been modelled before in the Baltic Shield (Cassel and Fuchs, 1979; Lund, 1979). If these three diffracting features form part of the same structure, they may be related to the northern wall of the trench-like feature in Model A.

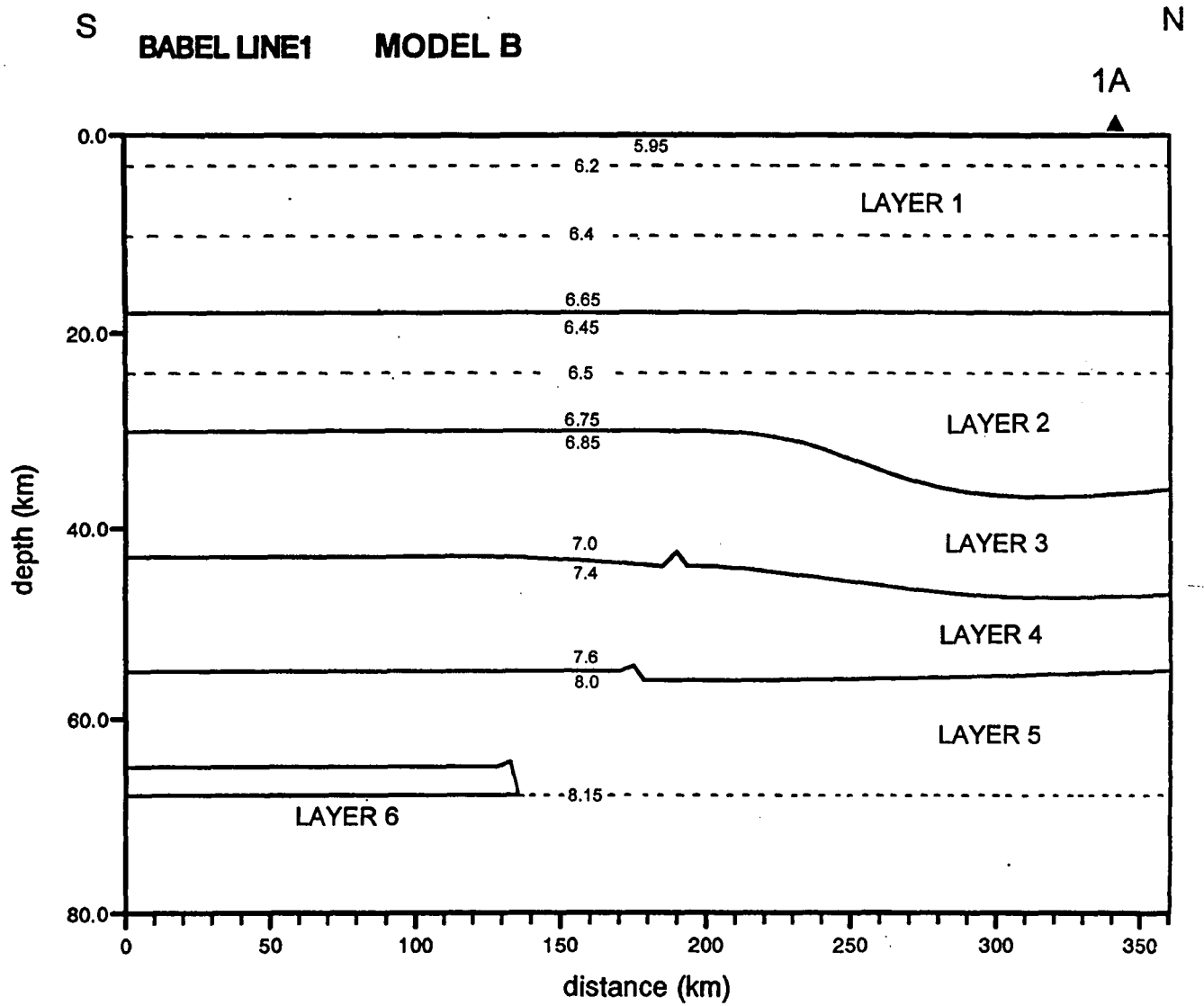


Fig. 7.24 Model B: Velocity model for off-line station 1A.

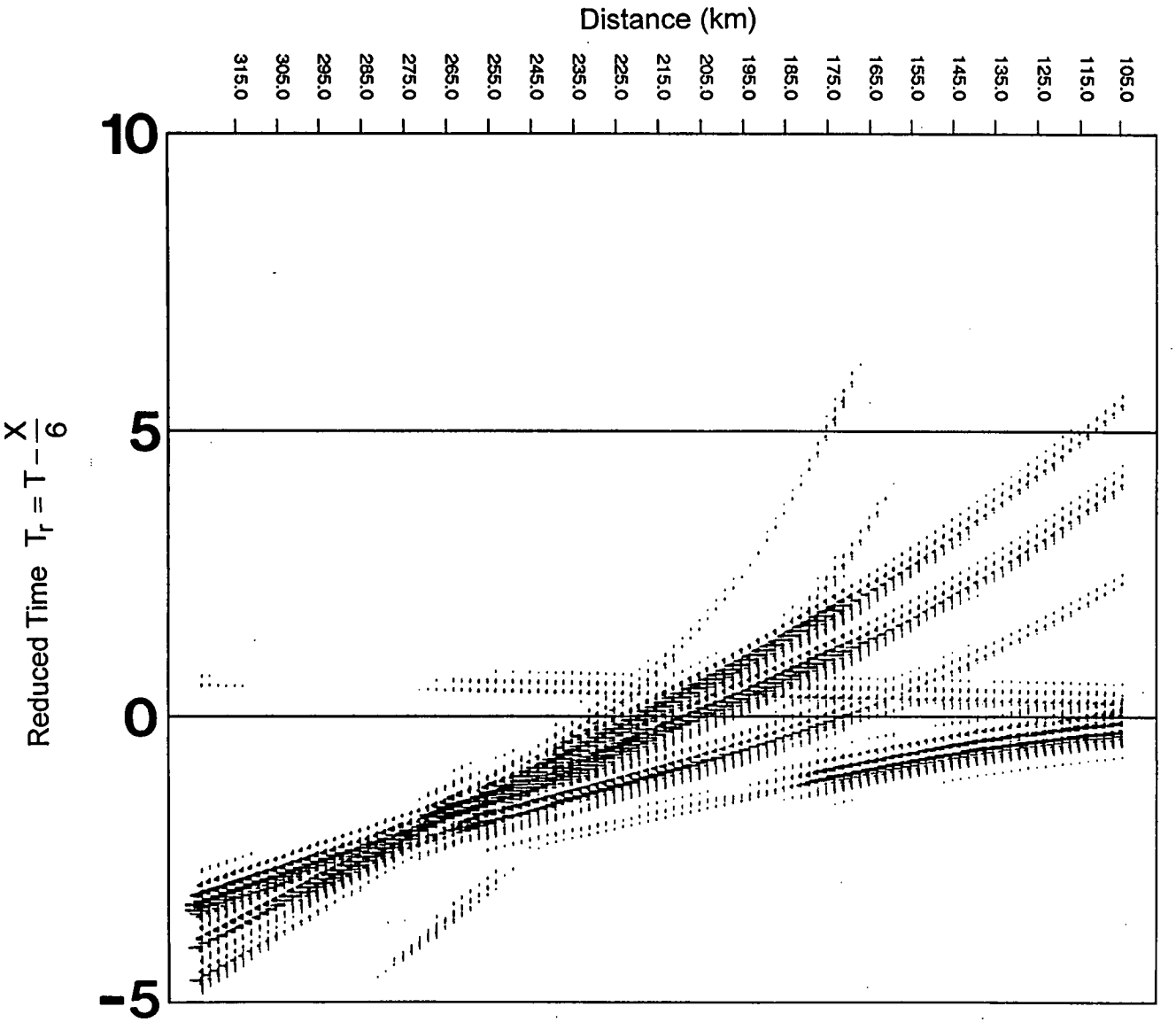


Fig. 7.25 Synthetic section generated for station 1A from Model B.

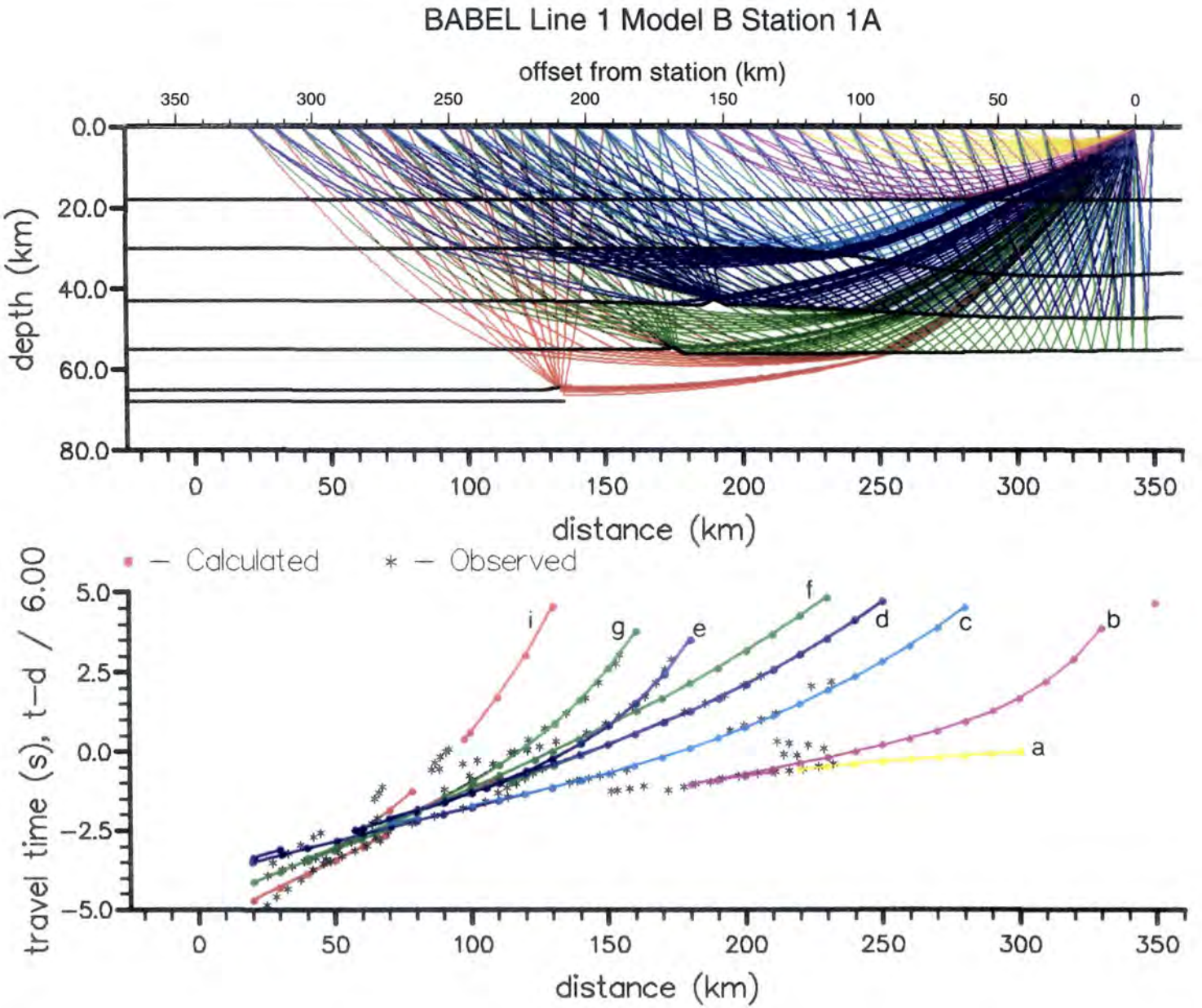


Fig. 7.26 Raypaths and travel time curves for Model B, station 1A.

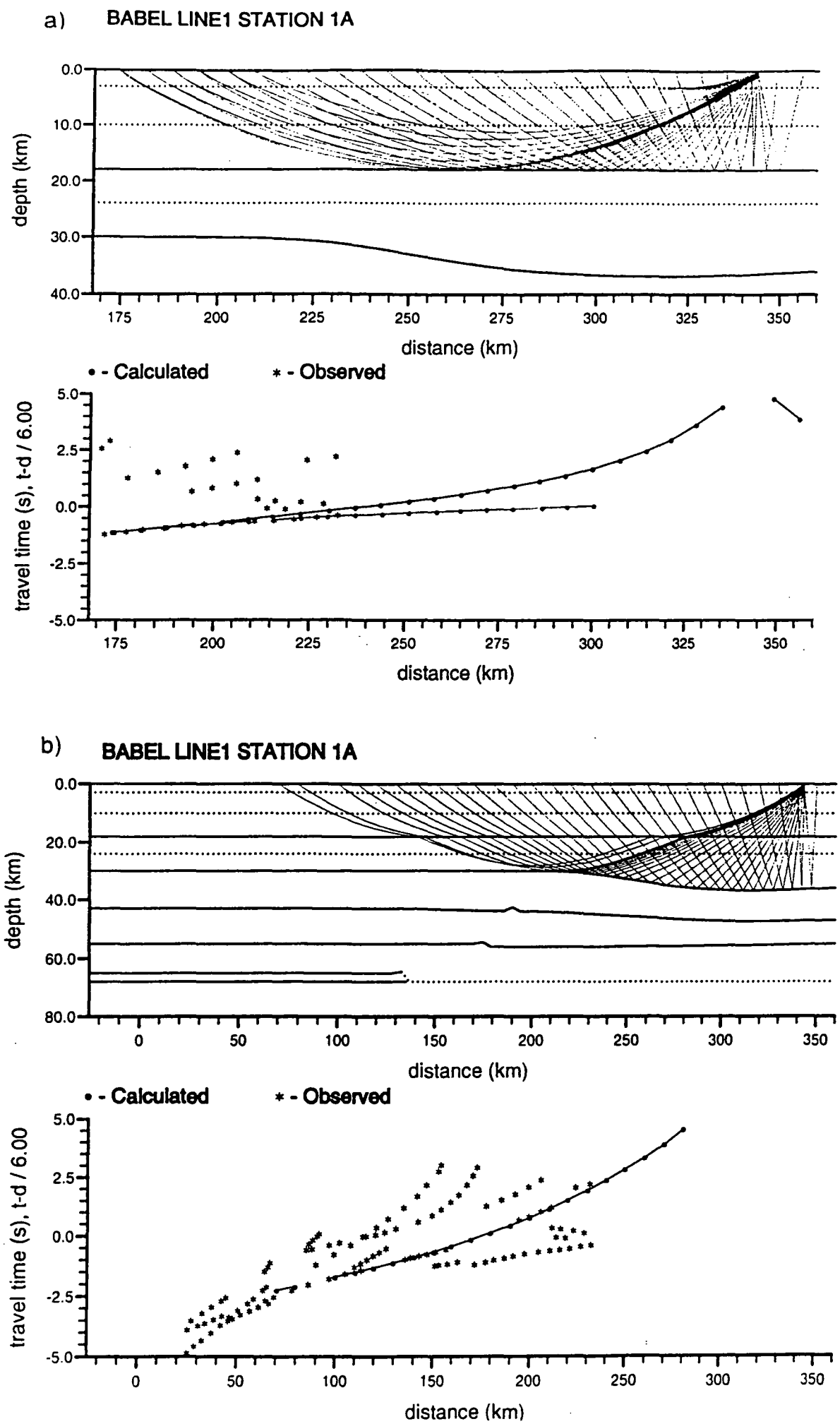


Fig. 7.27 Breakdown of raypaths and travel time curves for Model B, station 1A.

a) Layer 1

b) Layer 2

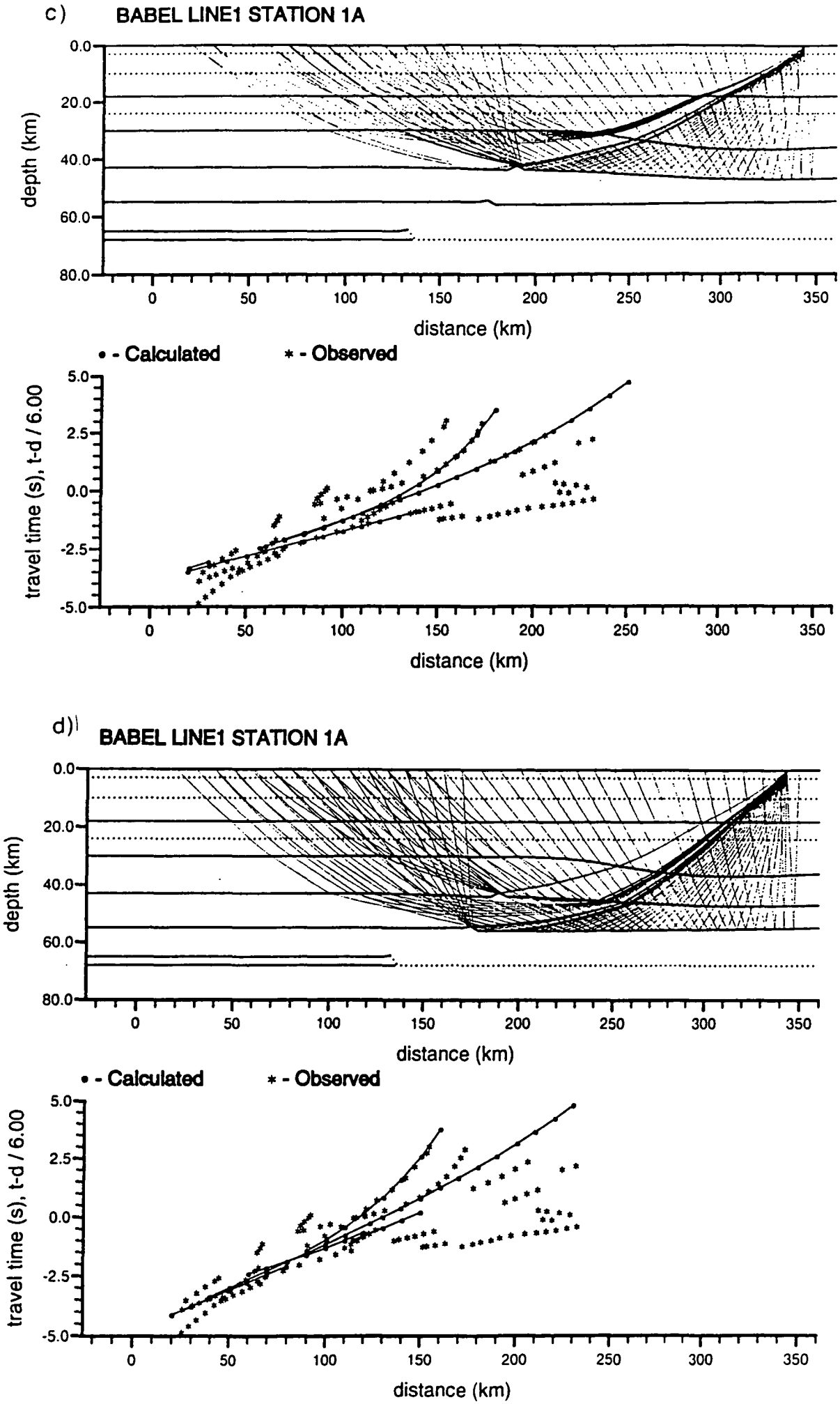


Fig. 7.27(cont.) Breakdown of raypaths and travel time curves for station 1A.

c) Layer 3

d) Layer 4

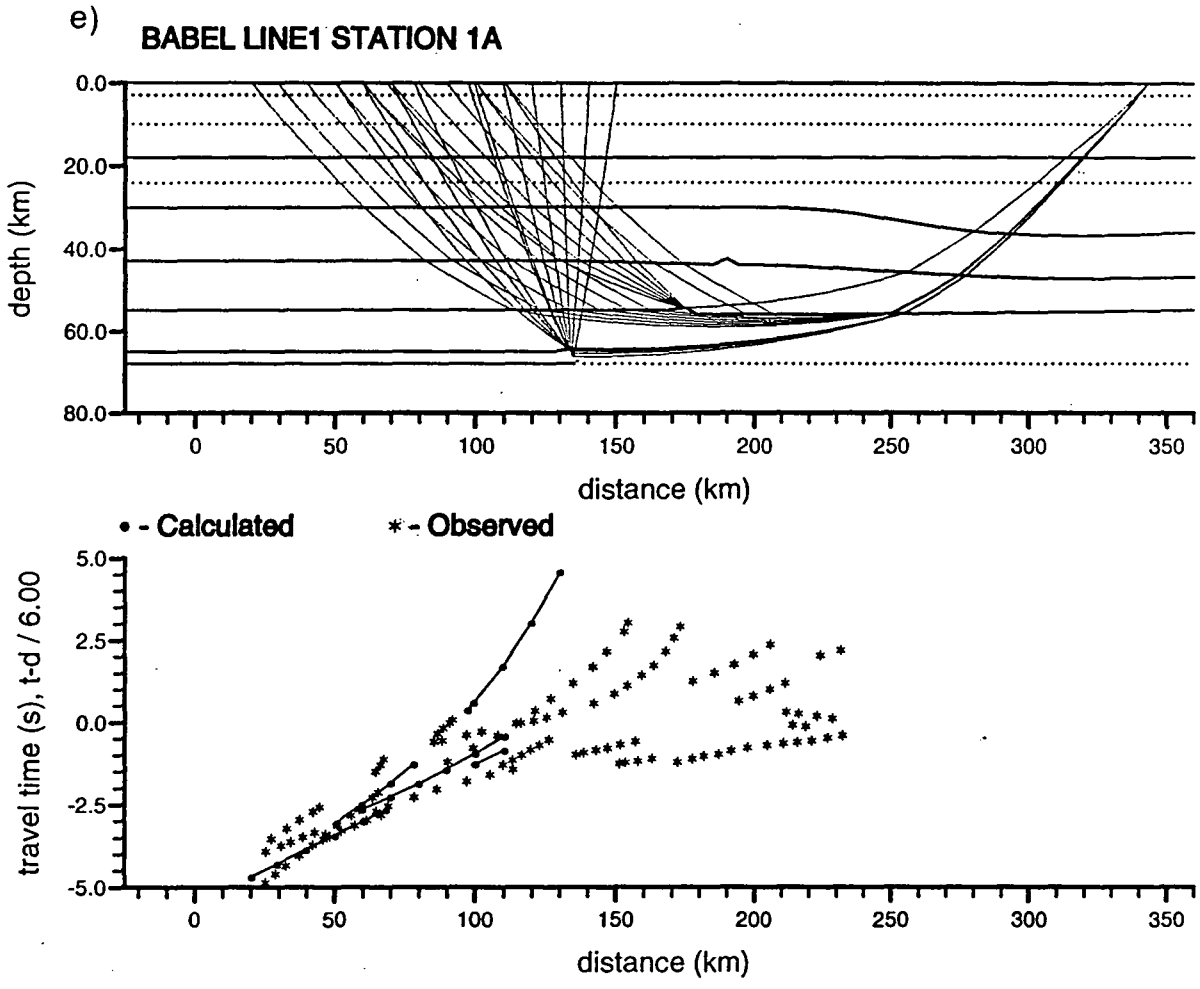


Fig. 7.27(cont.) Breakdown of raypaths and travel time curves for station 1A.

e) Layer 5

7.3.4 Model C : Station 5A

The model for the southern station 5A bears greater resemblance to Model A (Figs. 7.28 to 7.31). The upper crust (layer 1) is again not defined by the data. The early travel time of the first arrival with an apparent velocity of around 6 km s^{-1}, however, proved difficult to model. This arrival appears to have a varying apparent velocity and is interpreted as being made up of a higher velocity arrival (*b*) from a diving ray in a high velocity layer between 5 and 9 km depth (layer 3) followed by a lower velocity reflection (*a*) from the boundary between layers 2 and 3 (Fig. 7.31a). The high velocity layer 3 pinches out toward the centre of the model. The middle crust is fairly featureless with only one reflecting interface between layers 4 and 5 which dips from 25 km depth at the southern end to 31 km at the centre of the model and is defined by arrival *c* (Fig. 7.31b).

From the lower crust, again there are two strong reflections (*d* and *e*), the earlier of which is apparent only at shorter offsets (*d*, Fig. 7.31c) whilst the later is seen at longer offsets (*e*, Fig. 7.31d). Thus the first reflector is cut away to form a trench, as in Model A. Here, however, the northern wall is imaged from the south (arrival *f*). The upper mantle velocity is found from arrival *g* to be 8.1 km s^{-1} (Fig. 7.31e).

In theory, gravity modelling could be used to constrain Models B and C further, as for Model A. However, since both models are derived from off-line data, the interfaces at depth do not lie in the same vertical plane so 2D gravity modelling is not really appropriate.

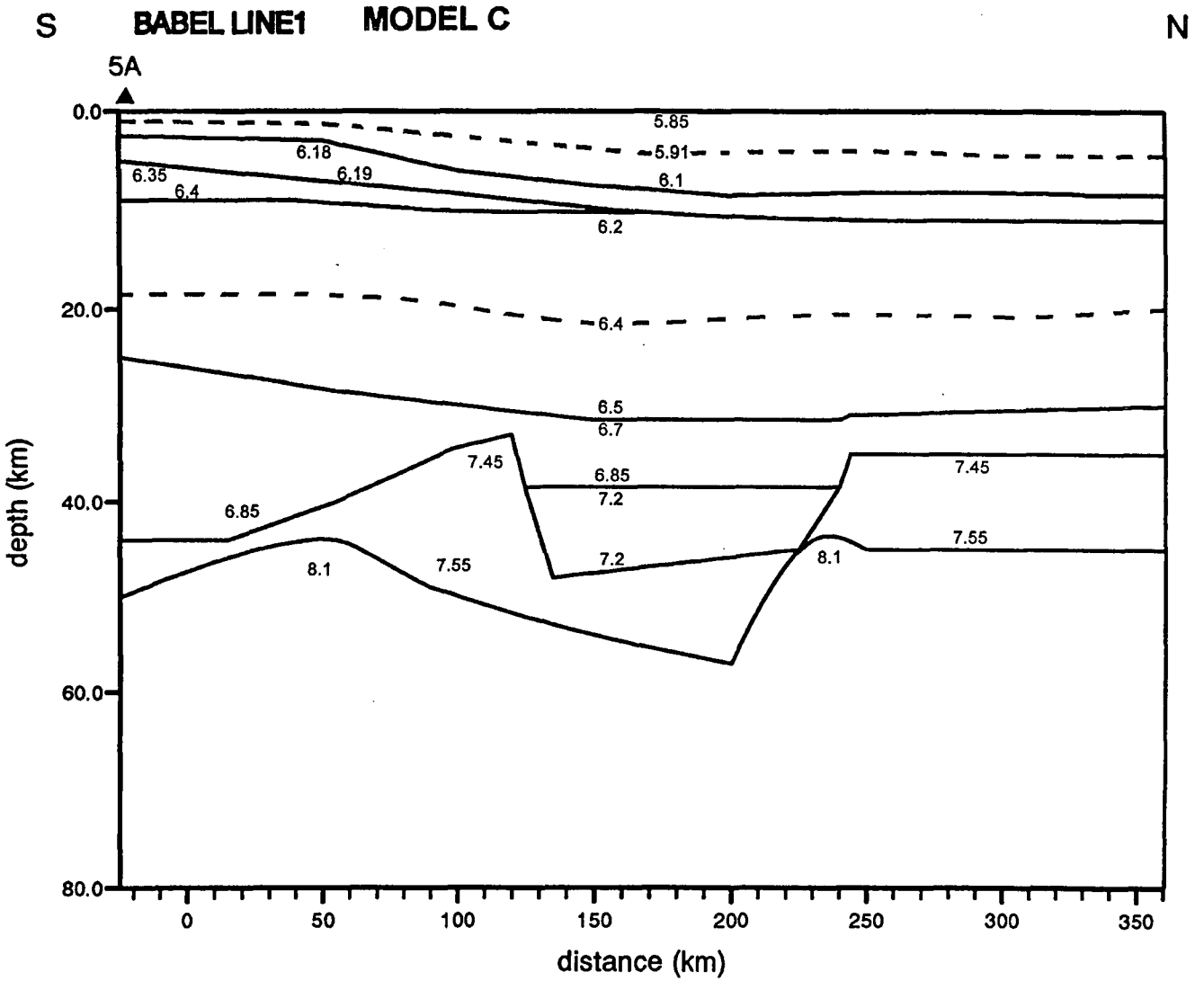


Fig. 7.28 Model C: Velocity model for off-line station 5A.

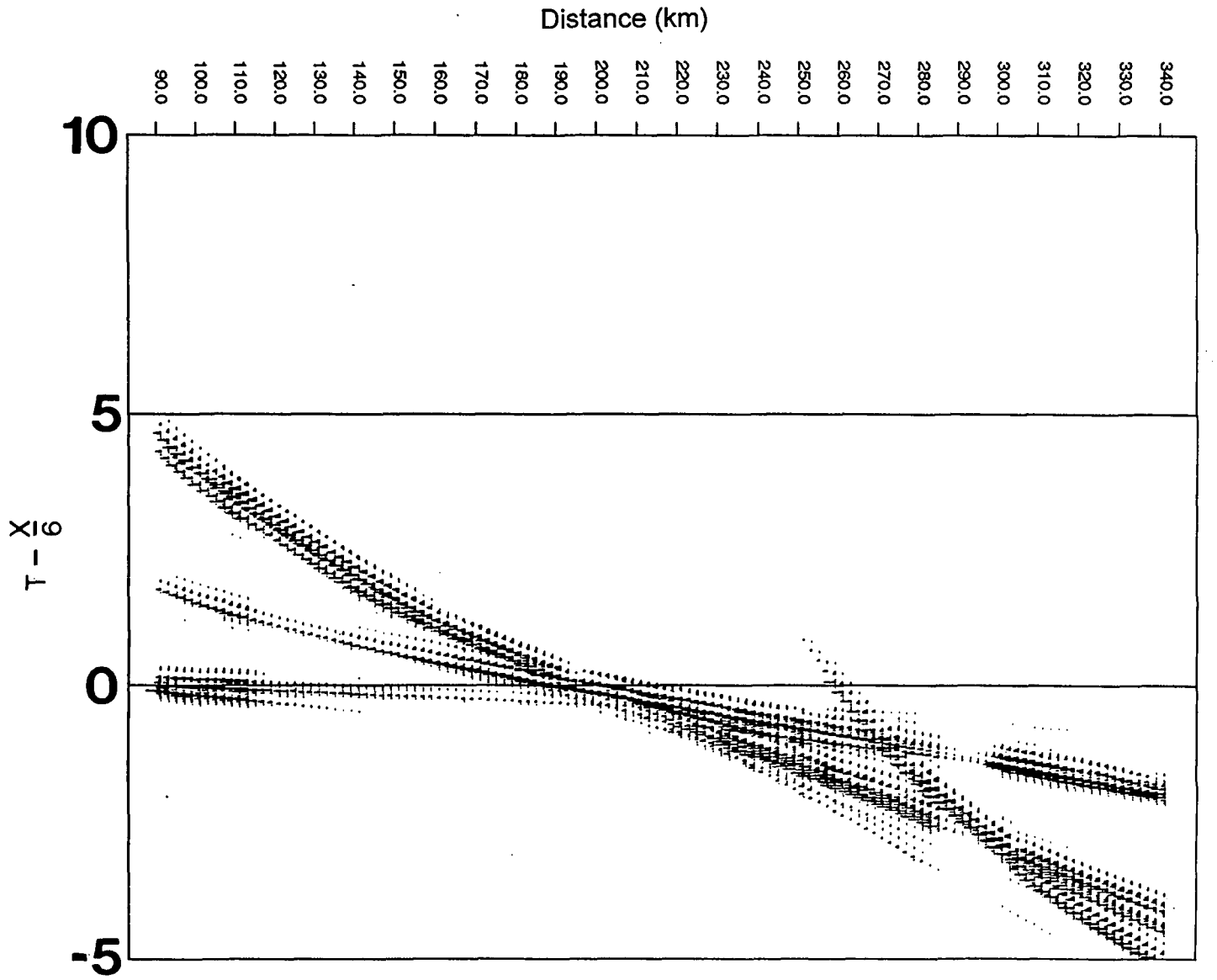


Fig. 7.29 Synthetic section generated for station 5A from Model C.

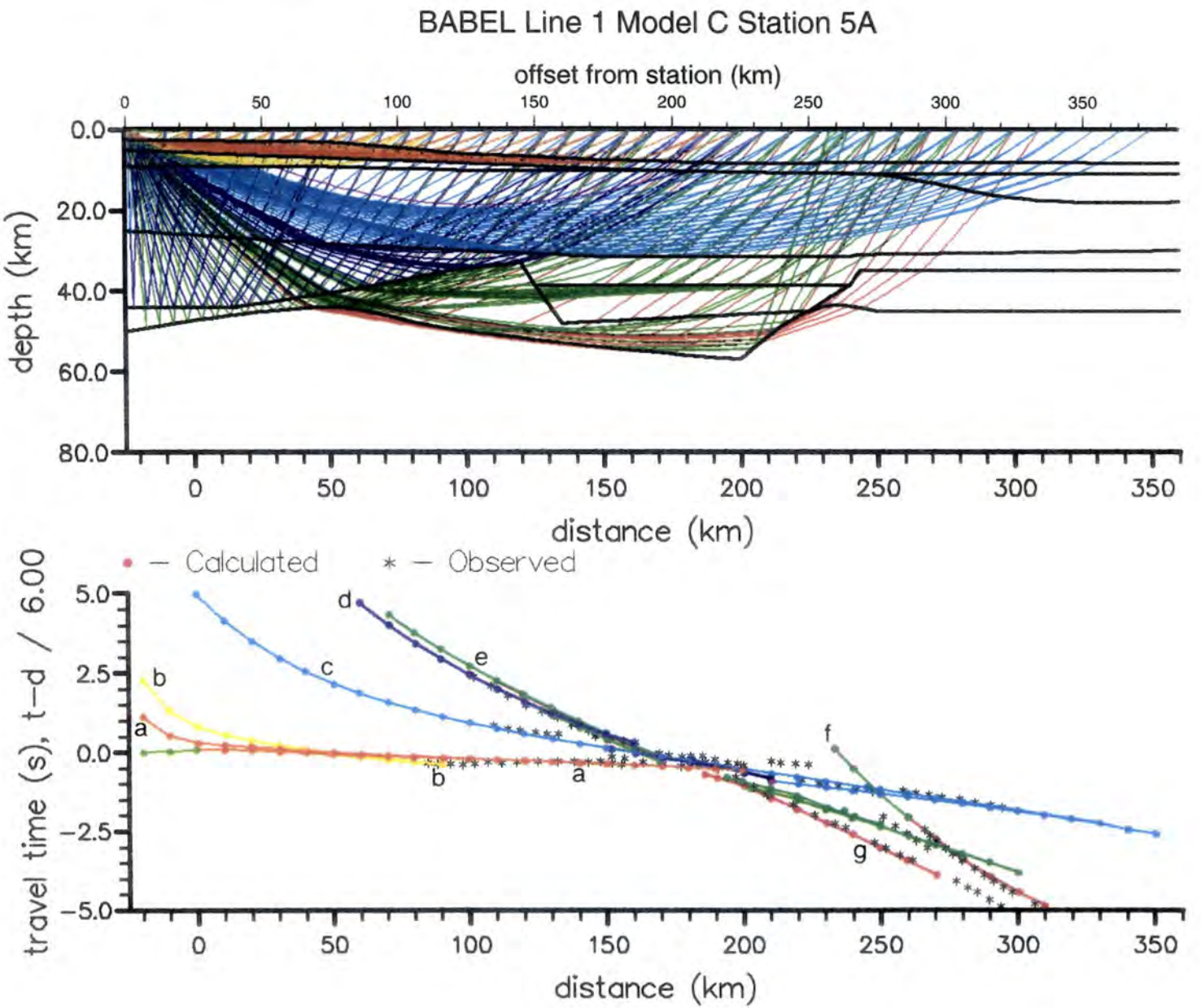
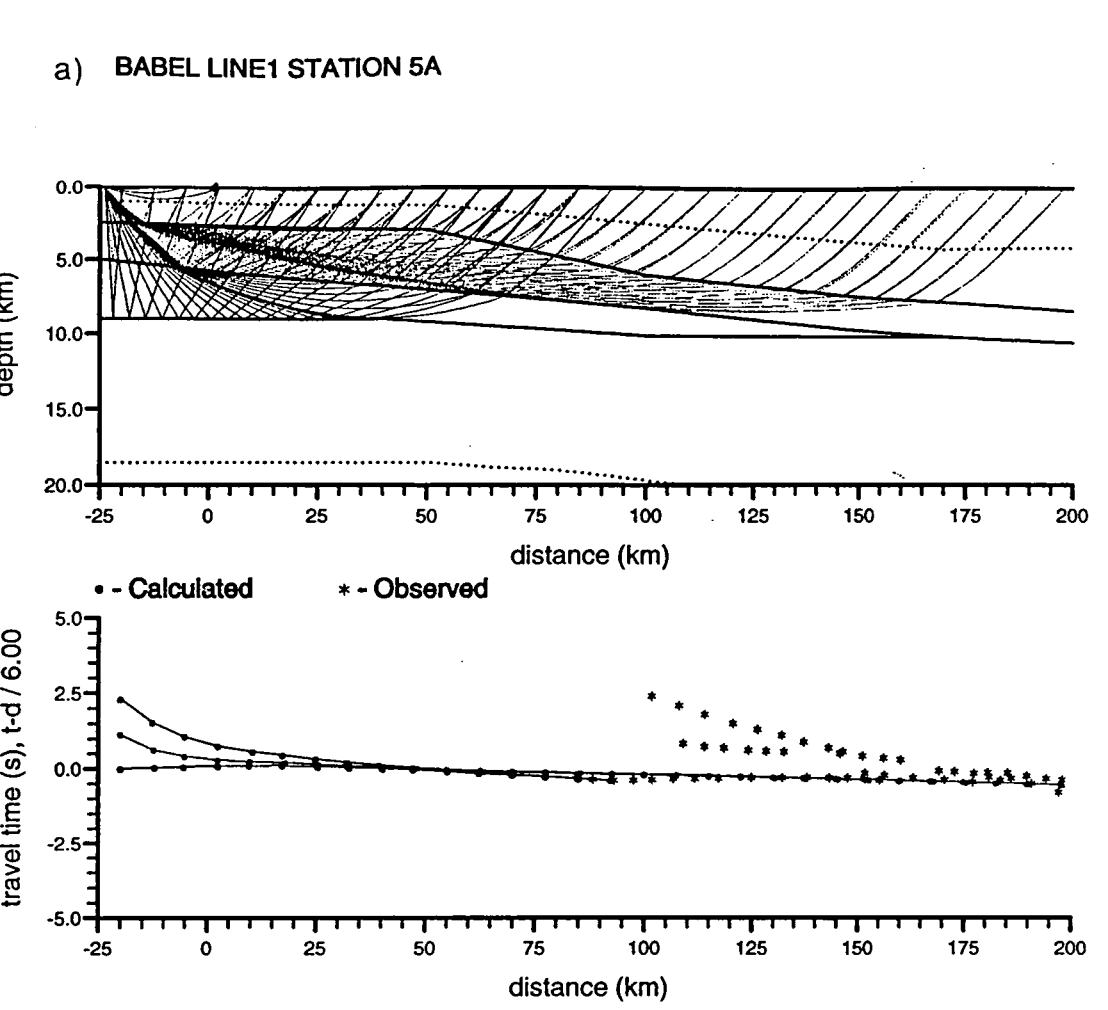


Fig. 7.30 Raypaths and travel time curves for Model C, station 5A.



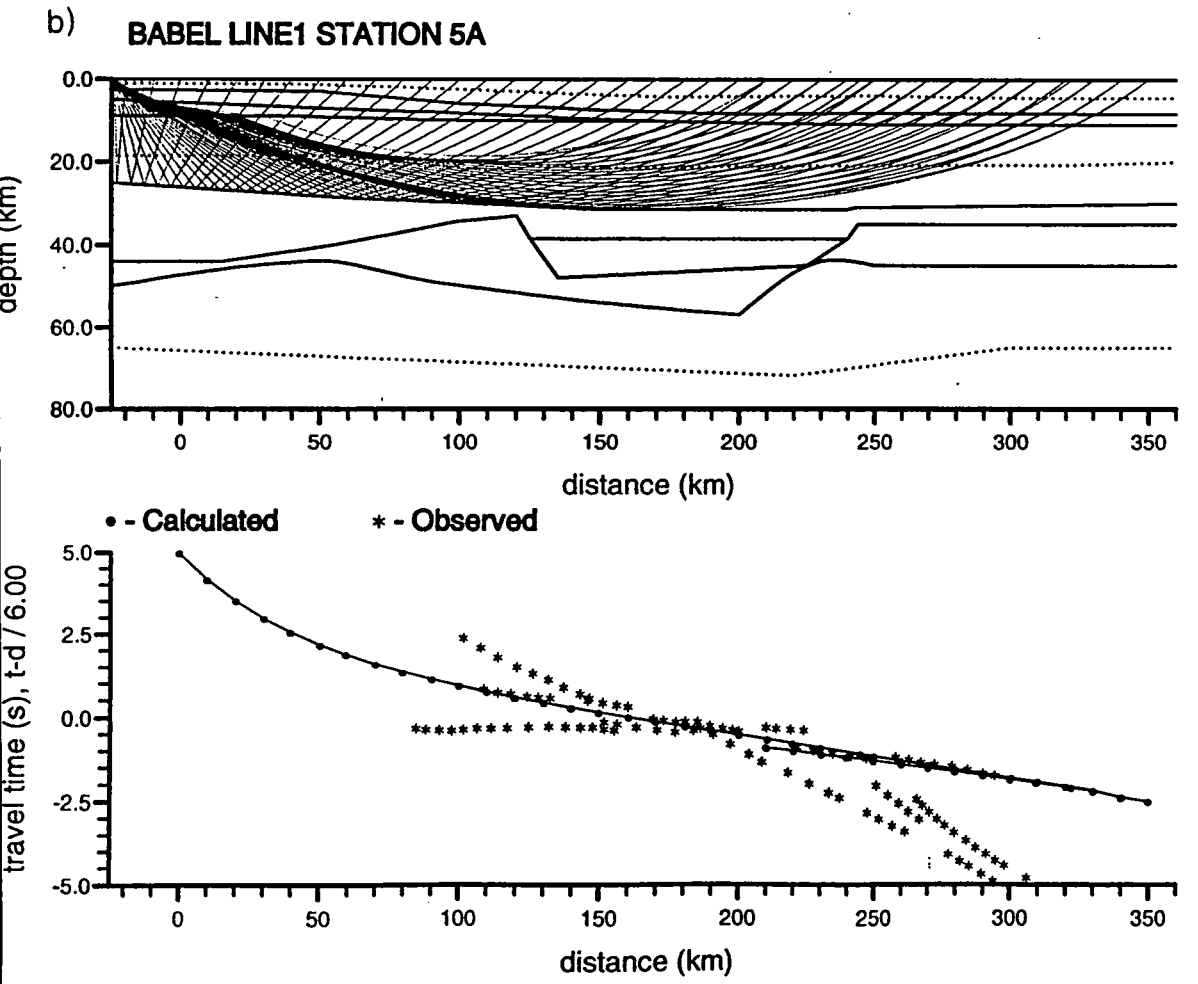
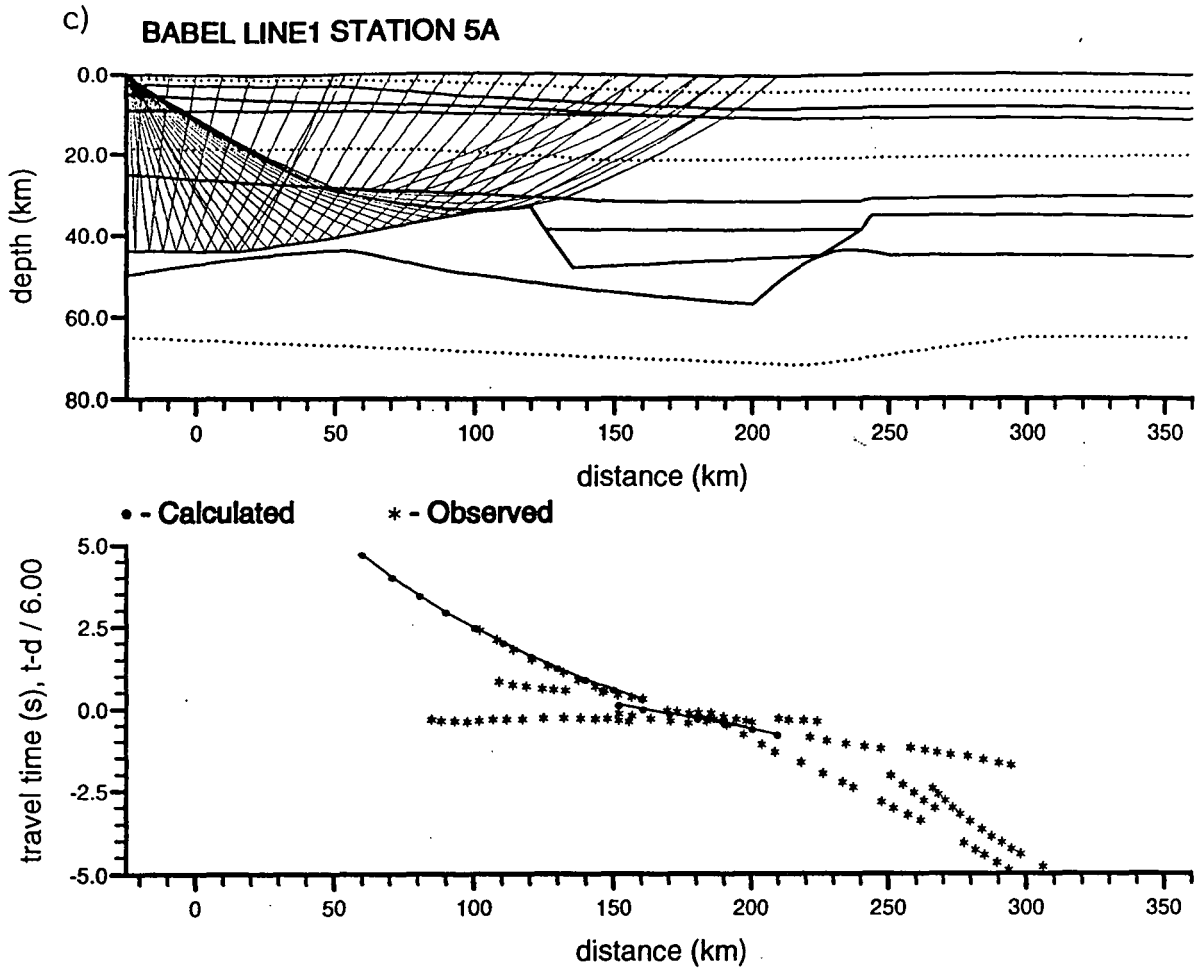


Fig. 7.31 Breakdown of raypaths and travel time curves for Model C, station 5A.

a) Layers 1, 2 and 3

b) Layer 4



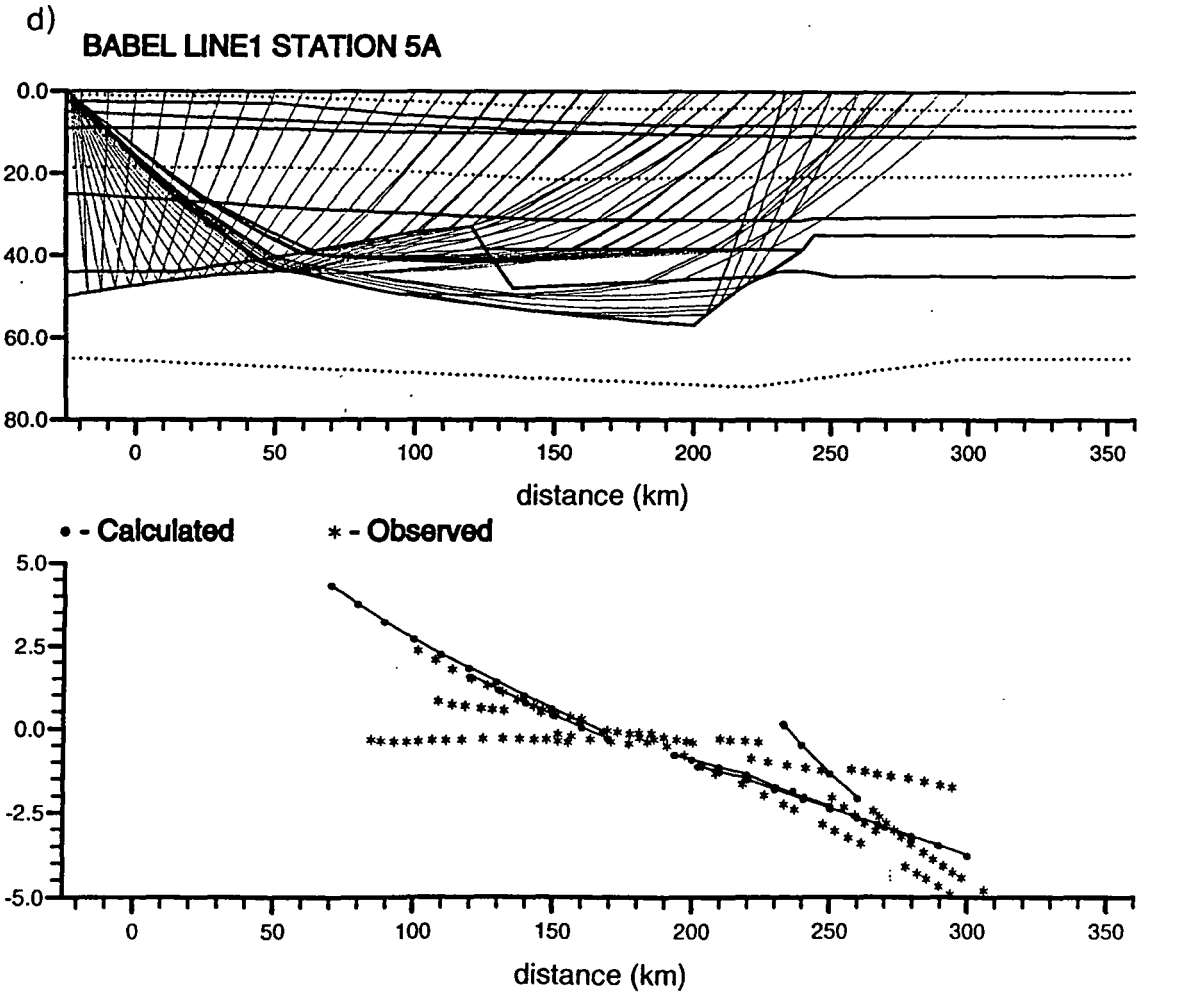


Fig. 7.31(cont.) Breakdown of raypaths and travel time curves for station 1A.

c) Layer 5

d) Layer 7

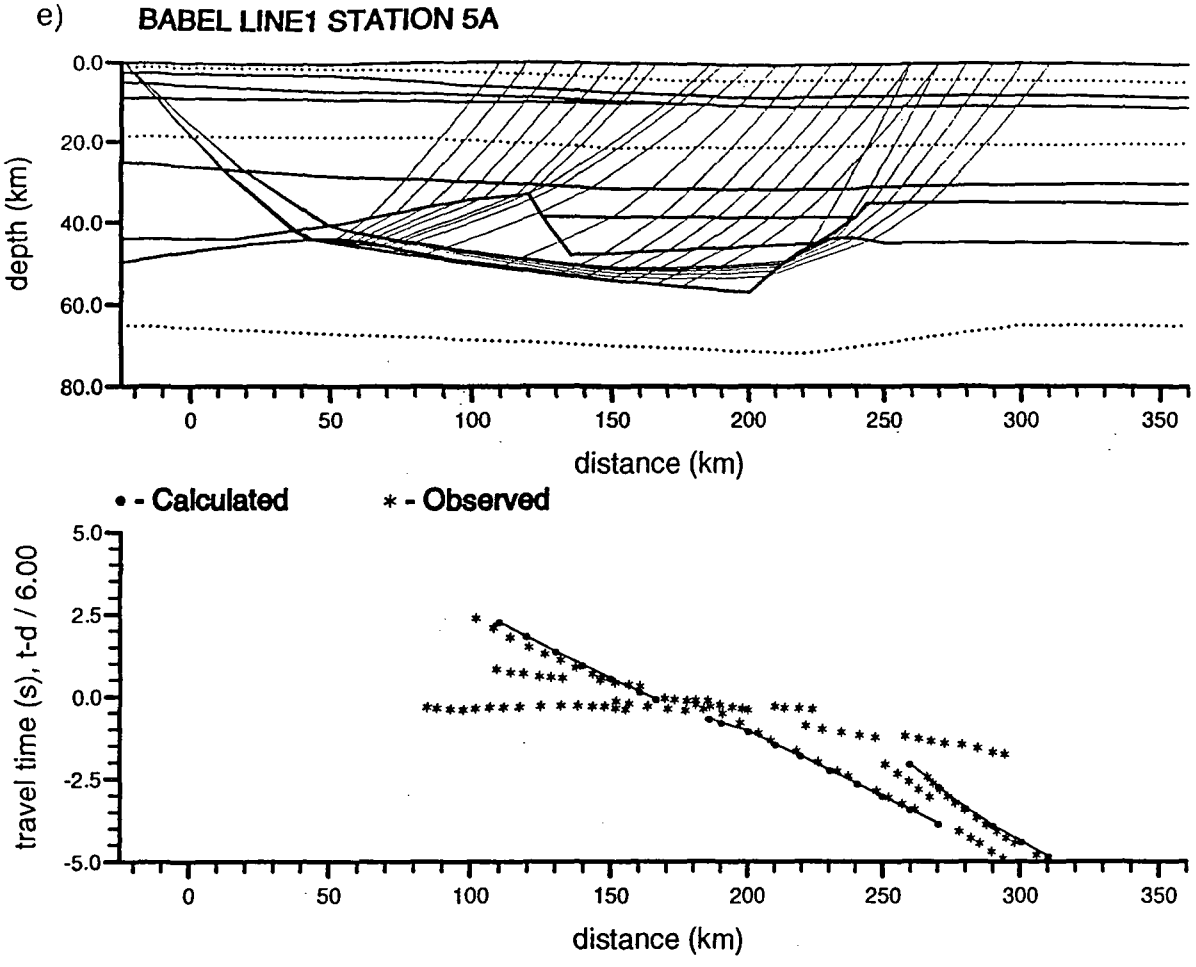


Fig. 7.31(cont.) Breakdown of raypaths and travel time curves for station 1A.

e) Layer 8

7.4 Accuracy of the model

The major sources of error in the model arose from the shot and receiver locations, the picking of arrivals, and the fitting of calculated travel time curves to digitised, picked arrivals. As stated earlier, the shot location error was taken to be ±50 m and that of the receiver locations ±100 m. The arrivals could be picked and digitised to an accuracy of ±10 ms (~$\pm^1/_4$ wavelength) and the modelled travel time curves fitted to a similar accuracy. The total resultant error in the model itself is harder to assess. The ray diagrams show that large regions are poorly covered and some not at all (Figs. 7.20, 7.21, 7.26 and 7.31). Few diving rays pass through the lower crust so the velocities here are theoretically not well constrained. However, the reflected rays from the lower crustal boundaries, especially divergent reflections from curved boundaries, give velocity information at large offsets similar to diving rays. This provides a certain level of constraint on velocities.

In the upper crust, little velocity information is known for the extremes of the model, or for the uppermost central part, where all the rays are near vertical. This lack of accurate velocity information for the uppermost crust leads to increased uncertainty in the rest of the model. In the deeper regions of the model where velocity is not well constrained, i.e. at the northern and southern extremes, trade-offs between velocity and depth are possible. On the other hand, many of the reflecting boundaries, especially those of high curvature (such as the mid- and lower-crustal diffracting points in Models A and B and the northern wall of the Moho trench in Model C), need only be displaced by tens of metres to destroy the fit of the travel time curve.

Gravity data provides further constraint to the model, especially in the upper crust from where most gravity anomalies arise. Two dimensional gravity modelling by itself cannot produce a uniquely determined model but, combined with the seismic model, it is invaluable in helping to determine the structure in parts of the model poorly covered by seismic rays and give a general idea of appropriate velocities from the density information. The resulting changes to the seismic model only go to demonstrate the non-uniqueness of travel time modelling alone, i.e. seismic data also fails to adequately constrain the detail of a model.

7.5 Three dimensional aspects of the model

The major problems in working with three dimensional data were the limitations of the modelling package (i.e. the model itself is limited to two dimensions) and the presentation of three dimensional information.

Although a 3D ray tracing package became available at a late stage during this project (ANRAY86, Gajewski and Psencik, 1986), it had several major drawbacks which made it unsuitable for modelling the BABEL Line 1 dataset. The main disadvantage was the severe limitation in the complexity of the model which could be defined, i.e. no merging boundaries, steps on interface surfaces or laterally varying velocities were allowed. In addition, modifications were necessary in order to model off-line stations which made the program extremely long to run even for simple models (the package was designed to model 3D anisotropic media along a profile) and the Gaussian Beam approximation was not available. These limitations,

coupled with the difficulty in defining and adjusting the model precluded the use of ANRAY86 in this study.

Section 7.2 described how the fan spread data from stations 1A and 5A were approximated to two dimensions and modelled as such. In order to get an idea of the three dimensional implications of these models, it was necessary to carry out the reverse process. Figs. 7.32a-c show the reflectors in Models A, B and C which were actually imaged in the data. For Models B and C, the modelled rays corresponding to the observed arrivals were traced back to their shotpoints. For each ray, the reflection point was marked on a plan view of the area at the correct distance from the receiver along a straight line between the shot and receiver. It was assumed that the rays lay entirely in the vertical plane containing the source and receiver, i.e. no out of plane reflections were occurring, and also that the velocity structure close to each off-line station did not vary greatly with azimuth. For Model A, the reflection points were all assumed to lie in the vertical plane of the profile. Figs. 7.33a-d show the loci of reflection points resulting from this transformation. The reflecting boundaries in the three models were roughly correlated into four groups of similar depth and velocity.

The uppermost crust was ignored for the three dimensional interpretation as it was not imaged by station 1A, though there is a good correlation between the boundary at the southern end of both Model A (6.30 - 6.35 km s^{-1}) and Model C (6.19 - 6.35 km s^{-1}) at ~7 km depth. Fig. 7.33a shows upper crustal boundaries across which the velocity decreases from between 6.55 and 6.7 km s^{-1} to between 6.35 and 6.55 km s^{-1}, from 6.4 to 6.2 km s^{-1}, and increases from between 6.35 and 6.55 km s^{-1} to between 6.55 and 6.6 km s^{-1}. These are too few and too far

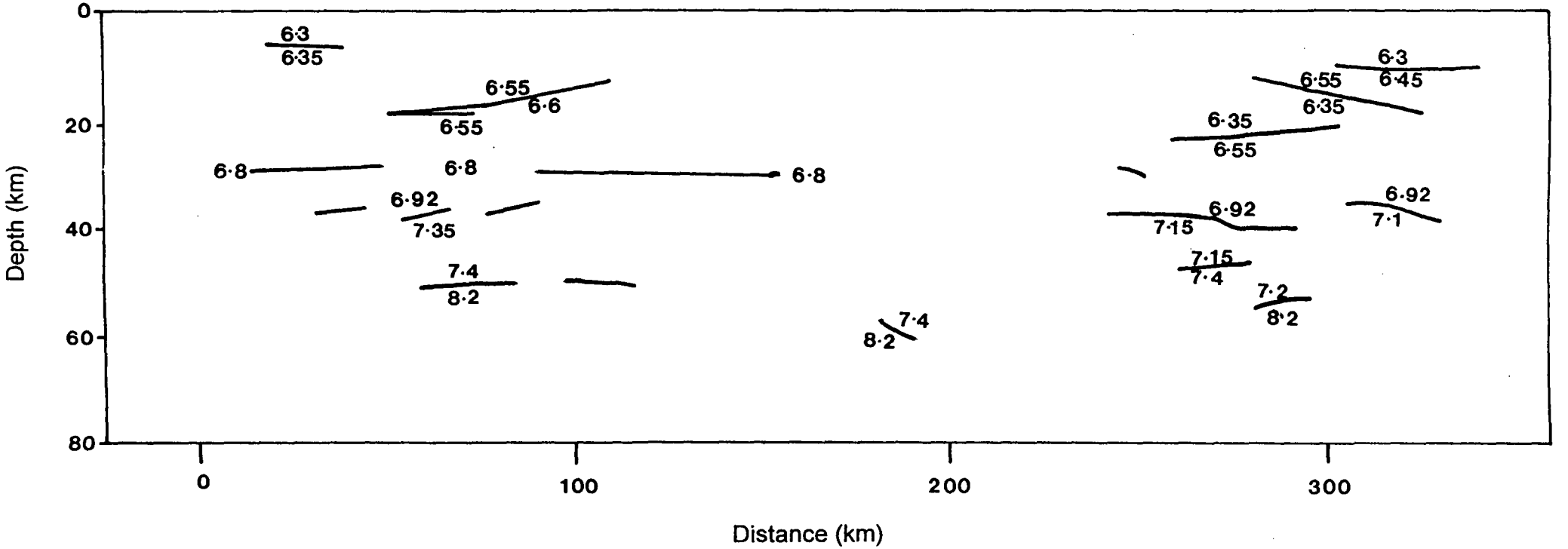


Fig. 7.32 a) Reflectors in Model A imaged in the wide angle data.

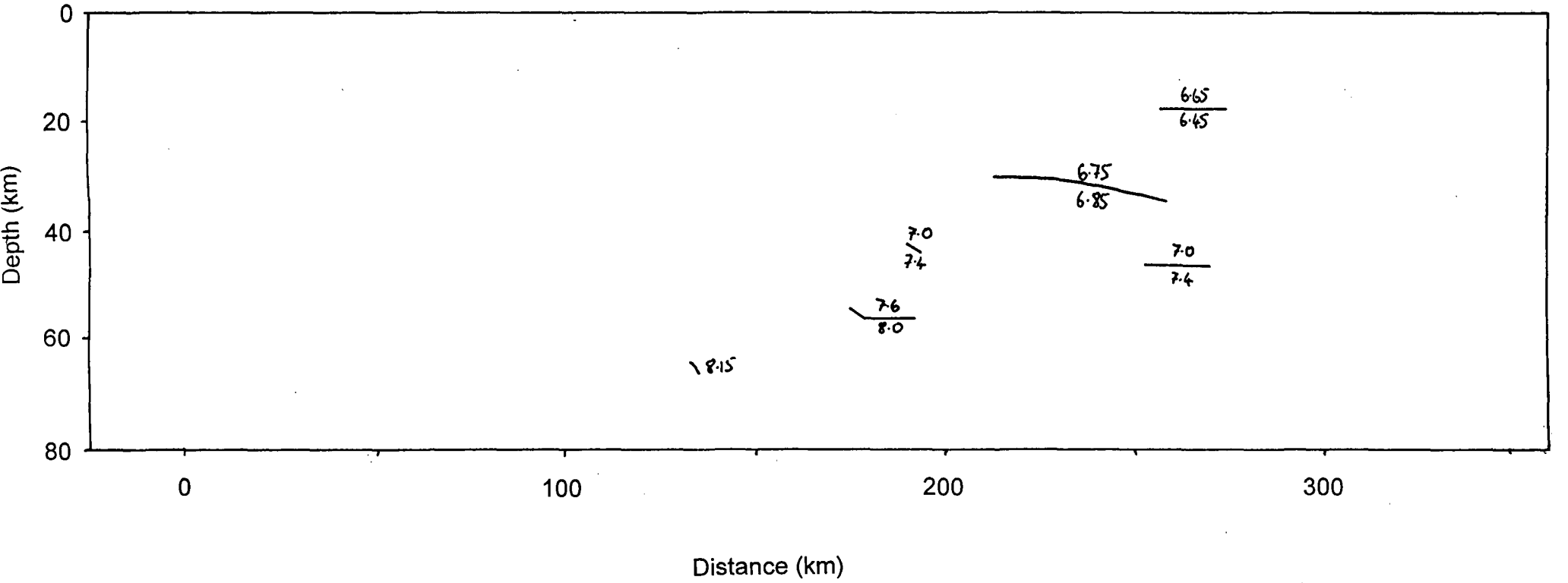


Fig. 7.32 b) Reflectors in Model B imaged in the wide angle data.

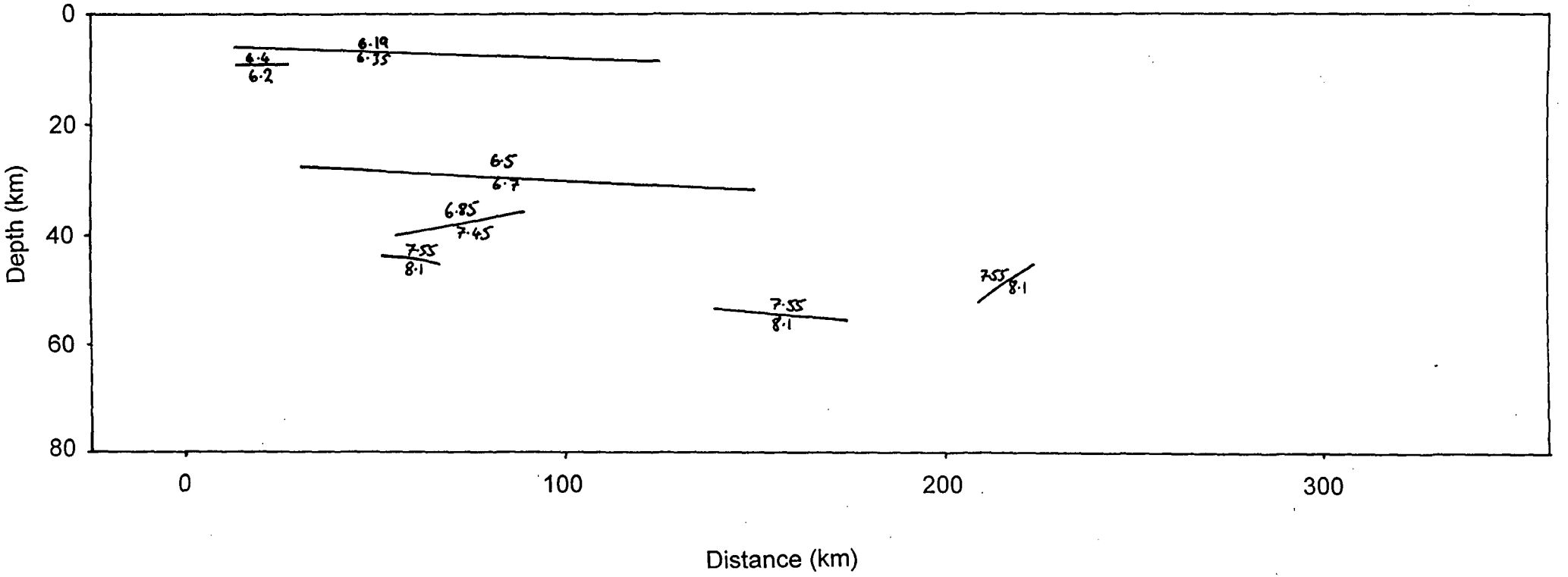


Fig. 7.32 c) Reflectors in Model C imaged in the wide angle data.

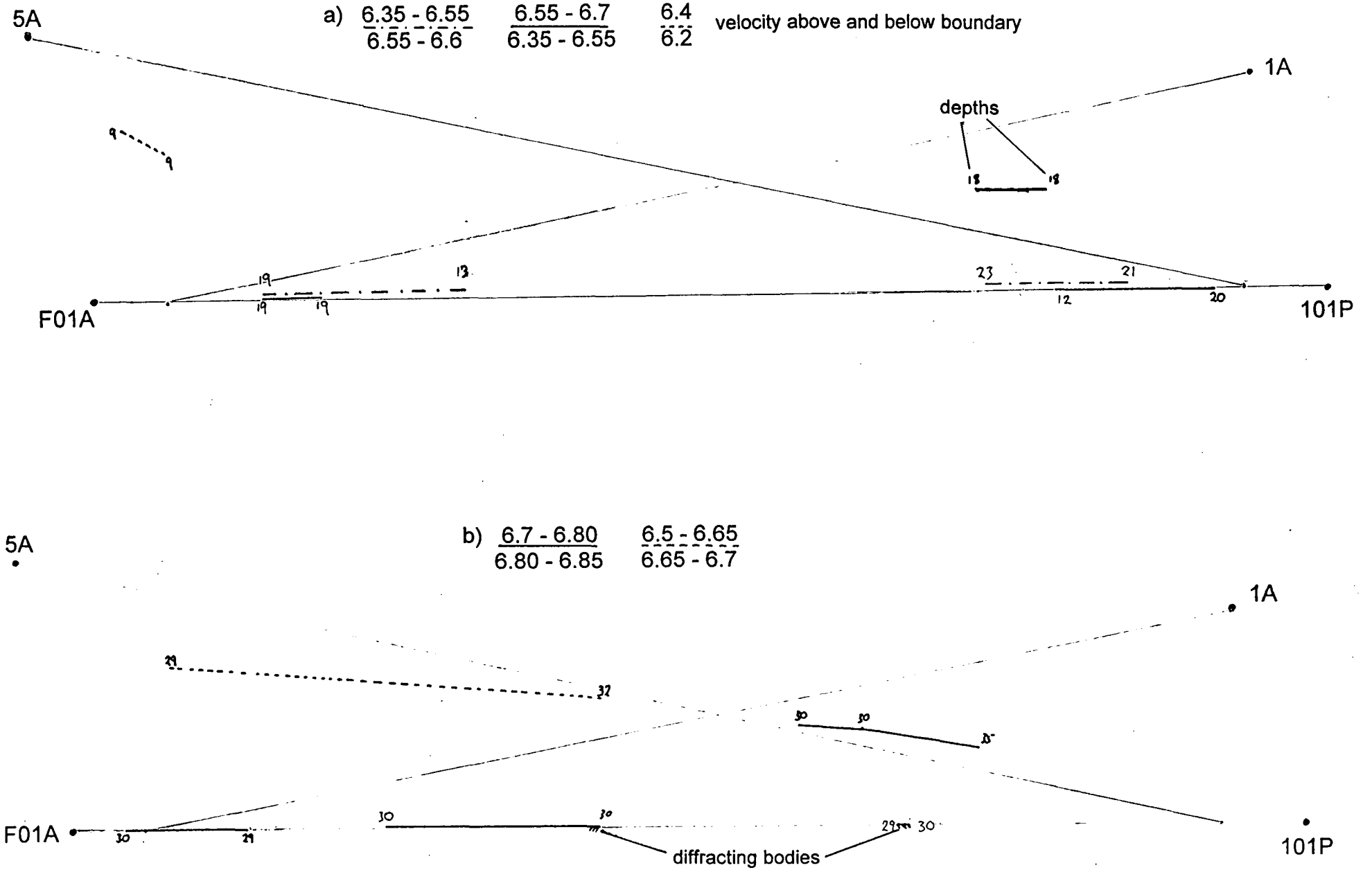
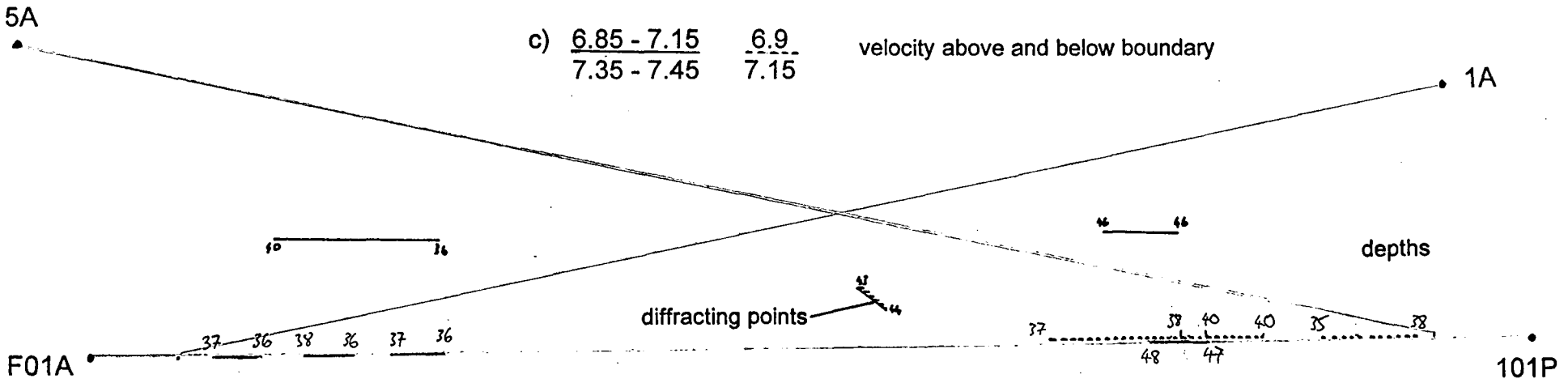
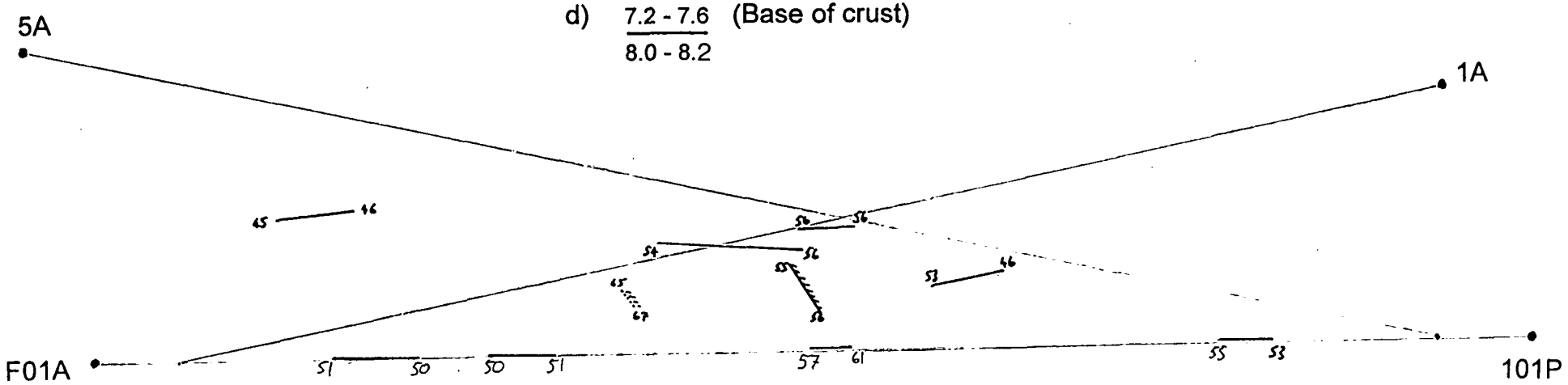


Fig. 7.33 3D aspects of the model: a plan view of the loci of reflection points in the three models.


5A
c) 6.85 - 7.15 / 7.35 - 7.45 6.9 / 7.15 velocity above and below boundary
1A
depths
diffracting points
F01A
101P


5A
d) 7.2 - 7.6 / 8.0 - 8.2 (Base of crust)
1A
F01A
101P

apart to make any meaningful correlation between them. Fig. 7.33b shows the boundaries between 6.5 km s^{-1} and 6.85 km s^{-1}, including the thin reflecting layers and diffracting bodies in Model A. It is obvious from the diagram that there is a wide spread discontinuous reflector of some sort at around 30 km depth.

The boundaries between mid and lower crust are grouped in Fig. 7.33c. There appears to be some correlation between the short dipping reflectors at the southern end of Model A and that in Model C. The layer of velocity 7.1-7.2 km s^{-1} at the northern end of Model A does not appear to extend westwards as it is not seen in Model B. The boundary separating velocities ~7.0 and ~7.4 km s^{-1} is deeper to the north.

Much more detail can be seen in the base of the crust (Fig. 7.33d). The boundary dips from all directions towards a point roughly in the centre of the region. It is not clear whether this region of thickened crust extends either to the east or to the northeast of Line 1 but the evidence from stations 1A and 5A seems to indicate that the trend is approximately NE-SW and the dip is laterally limited to the west. The deep diffracting body seen in Model B does not appear to be related to any other feature in the three models.

If the structures controlling gravity in Model A (i.e. the high density upper crustal block and the Moho depression) could be linked with similar features in Models B and C, this could assist in defining the 3D structure. However, as mentioned above (section 7.3.4), 2D gravity modelling would be difficult for the off-line models and the upper crustal structures which give rise to much of the observed gravity anomaly are not imaged in these models due to the long offsets involved. Having said this, the NE-SW trend

of both the positive and negative gravity anomalies (Fig. 3.5) agrees with the trend of the Moho depression observed in the seismic models above.

7.6 Normal Incidence

The BEAM87 modelling package, designed for wide angle work, was modified by P.A.Matthews for normal incidence. Fig. 7.34 shows a normal incidence synthetic section generated by this for the in-line wide angle model. A discussion of the fit of the synthetic to the real data, together with an overlaid diagram, is given in the next chapter.

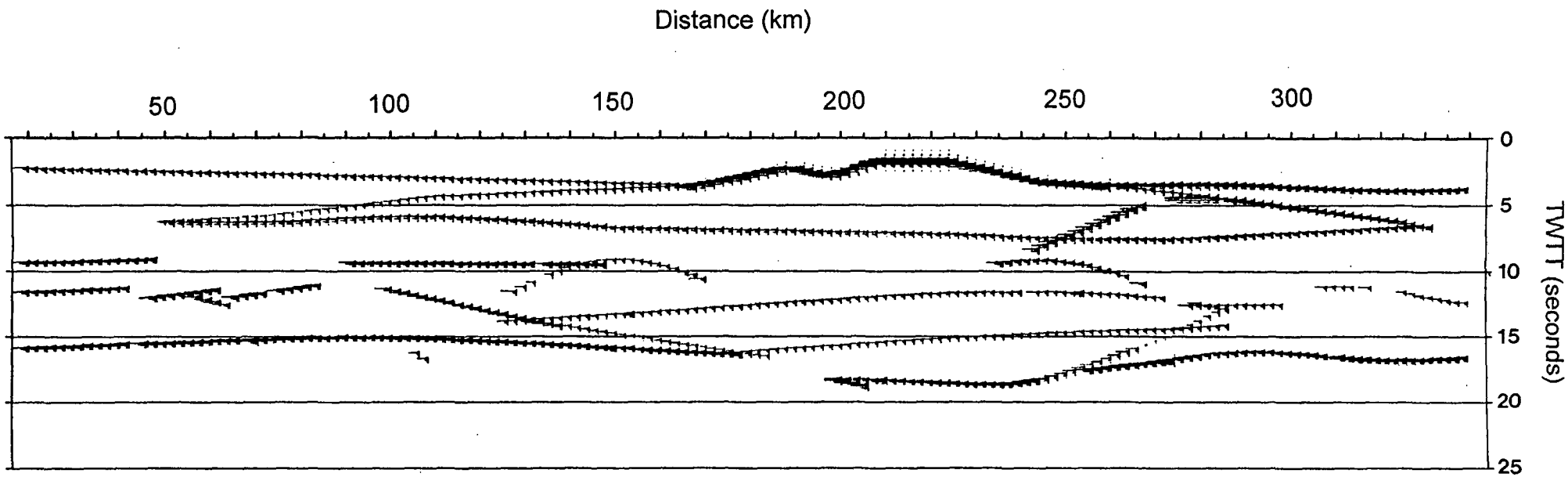


Fig. 7.34 Synthetic normal incidence section generated for the in-line model. True relative amplitude - no a.g.c. applied.

8. Discussion and Conclusions

8.1 Suitability of modelling software

The most obvious thing that can be said about the crustal models resulting from this study is that the ray theory approximation used is not really suitable for modelling such strong lateral velocity and structural variations.

Ray tracing packages such as BEAM87 are designed for modelling a horizontally layered crust with only small lateral variations. This is a reasonable assumption for traditional, low resolution refraction and wide angle reflection surveys in which continuous arrivals are traced over large distances. Each such arrival is considered to arise from a single continuous layer or boundary.

Modern high resolution datasets such as BABEL Line 1 reveal that in many instances this kind of interpretation is not accurate. Short length broken arrivals and large amplitude variations with offset appear to be more usual. The normal incidence data also show little indication of horizontal layering, nor does the geological history of the Svecofennides support this idea. Episodic igneous intrusions are thought to be part of the process of cratonisation (Nelson, 1991) and this would distort the original reflectivity patterns. There is plenty of evidence at the surface for such intrusions of various ages and sizes in the Svecofennides.

A modelling package designed for horizontal layers will bias the resultant model towards horizontal layers and these will obviously be employed in any part of the model with little or no data coverage. Although

selection and picking of arrivals at the interpretation stage may have a greater effect on the final outcome, as suggested by Ansorge et al (1982), the initial assumption of horizontal layering undoubtedly affects the interpretation. This was avoided as much as possible in the interpretation of the BABEL Line 1 model by picking only coherent sections of arrivals and not attempting to join them together at this stage.

In addition to the validity of the horizontal layer assumption, there are also other limitations of the modelling package. The method of defining the model employed here (i.e. linear vertical interpolation of velocities between isovelocity interfaces) means that the vertical velocity gradient in a layer which pinches out can become unreasonably large. Also, the velocity field in the vicinity of a strong fluctuation in a boundary becomes difficult to predict.

True diffractors cannot be modelled, only approximated by surfaces of high curvature. This may, in fact, be a more accurate representation of geological features. However, the Gaussian beam method does not give accurate amplitudes for reflections from highly curved interfaces so diffraction amplitudes cannot be used to constrain the model to any great extent.

Despite these drawbacks, the wide angle synthetic sections produced from the models are beginning to reflect the amplitude variations seen in the observed data.

Nearly all the seismic models in the Baltic Shield region were produced using BEAM87 or its predecessor SEIS83. Thus it must be borne in mind that similarities between the models may be due in part to the limitations and bias of the modelling package.

8.2 Normal incidence data and synthetic section

8.2.1 Comparison of observed and synthetic sections

It is clear that the synthetic normal incidence section as a whole (Fig. 7.34) does not tie up well with its observed counterpart (Fig. 6.22). Fig. 8.1 shows the synthetic section overlain on the real data. Those reflections from boundaries observed at wide angle are highlighted (see also Fig. 8.3). Major discrepancies include the lack of continuous reflectors seen in the observed data, and the transparent zone in the central part of the record section. The highlighted reflections, however, seem to have a closer correspondence to the observed data.

At the southern end, the reflection from the observed part of the upper crustal boundary in the model (~2.5 s, 20-40 km) ties in well with a short length reflector in the normal incidence data though its continuation to the north is not apparent in the data. Interestingly, a stronger reflector just below this at 3 s has no counterpart in the synthetic section. The strong reflection from the top of the modelled high velocity body in the synthetic section (2-4 s, 170-230 km) is not seen in the data but there is a patch of high reflectivity here and evidence for a northward dipping reflector between 230 and 245 km. The reflection from the observed part of the upper crustal boundary in the model (~4 s) at the northern end ties in well with a short length reflector in the normal incidence data while to the south of this it seems to merge with the lower of the strong reflectors previously interpreted as dolerite sills.

The reflection at 7.5 s (260 - 310 km) ties in with an observed reflection which marks the lower limit of upper-crustal reflectivity. The

dipping reflector between these (5 - 7 s) also has its counterpart in the data, especially at the bottom, and at the top where it merges with one of the strong reflectors interpreted as dolerite sills. The reflections at the southern end of the synthetic section at ~5 and 6 s (50 - 110 km) also tie in with observed reflections, the lower of these again appearing to be the base of a reflective zone.

The strong horizontal reflector seen near the middle of the synthetic section at about 9 s, 90 - 145 km, does not appear as a continuous arrival in the normal incidence data but there is evidence for short length reflectors close to the diffracting endpoint of the modelled reflector (140 km) and to its southern end (90 - 105 km). A similar reflector at the southern end of the model (9 s, 20 - 45 km) is quite well imaged in the normal incident data.

The normal incidence data show that the actual situation in the mid- to lower crust at the southern end of the model is far more complex than the simple jagged reflector modelled would suggest. Many short length horizontal and dipping reflections are seen, some of which can be tied in with the synthetic reflections from the parts of the jagged reflector observed at wide angle (~12 s, 30 - 80 km). The reflections at the northern end at similar travel times correspond closely with the top of lower crustal reflectivity.

The deepest reflector at the southern end of the synthetic section (~15 s) does not agree in travel time or dip with the deepest major reflector in the observed section although there is a case for it corresponding to the base of reflectivity. Further to the north, between 100 and 180 km, the reflector at 15 - 16 s does correspond well to the base of reflectivity.

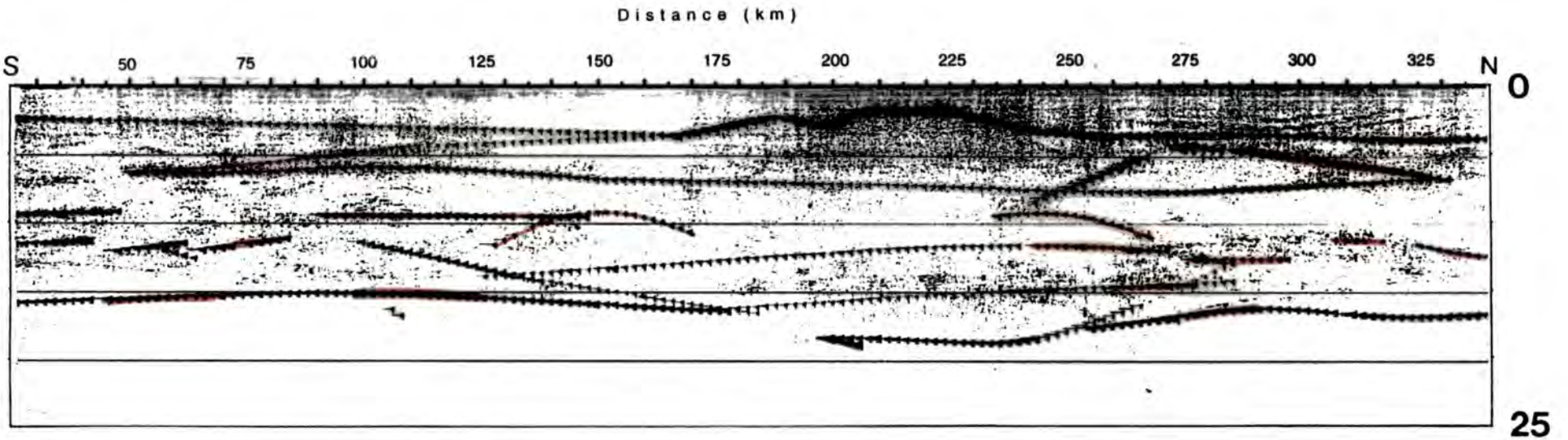


Fig. 8.1 Normal incidence synthetic section overlain on the real data. Reflections from boundaries observed at wide angle are highlighted. An a.g.c has been applied to the real data but not to the synthetic data.

At the northern end, the deepest boundary in the model can be tied in with a strong, short-length reflector in the normal incidence data which again marks the base of reflectivity (~16 s, 270 - 285 km), though the observed reflection has a slightly earlier arrival time than the synthetic one.

The only evidence for the modelled Moho depression is the continuation of reflectivity to deeper levels than the surrounding regions at the corresponding part of the observed section. No base to this is seen but the reflectivity gradually fades out. In addition, the base of crustal reflectivity on either side dips towards the position of the modelled depression.

Overall, it can be claimed that nearly all the synthetic reflections arising from the parts of modelled boundaries defined by wide angle reflections can be matched with observed normal incidence arrivals. However a great many more reflected arrivals are seen in the data which are absent from the synthetic section. The reasons for this are unclear. It may be that the short reflectors at the extremes of the dataset only appear in the wide angle data as low amplitude subcritical reflections which are not easily distinguished from multiple reverberations and background noise. Other reflectors may be of such short length and thickness that they are not seen as individual reflectors at wide angle but merely contribute to the complexity of other arrivals by scattering the energy.

8.2.2 Problems and assumptions made in the comparison

In comparing a normal incidence synthetic section generated from a model based on wide angle observations with the real normal incidence data, assumptions are made and several problems arise. It is assumed

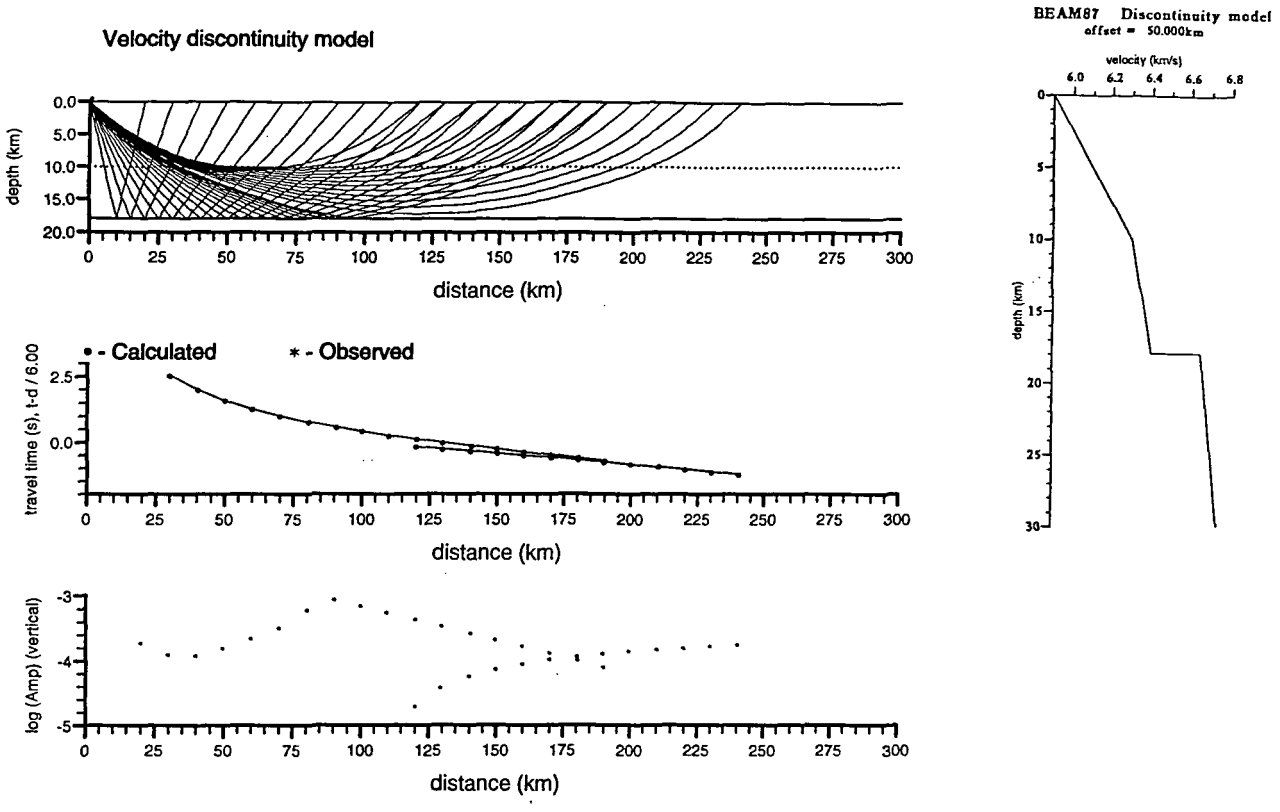
that features producing wide angle reflections will also produce normal incidence reflections and vice versa, i.e. wide angle and normal incidence reflections arise from the same geological features.

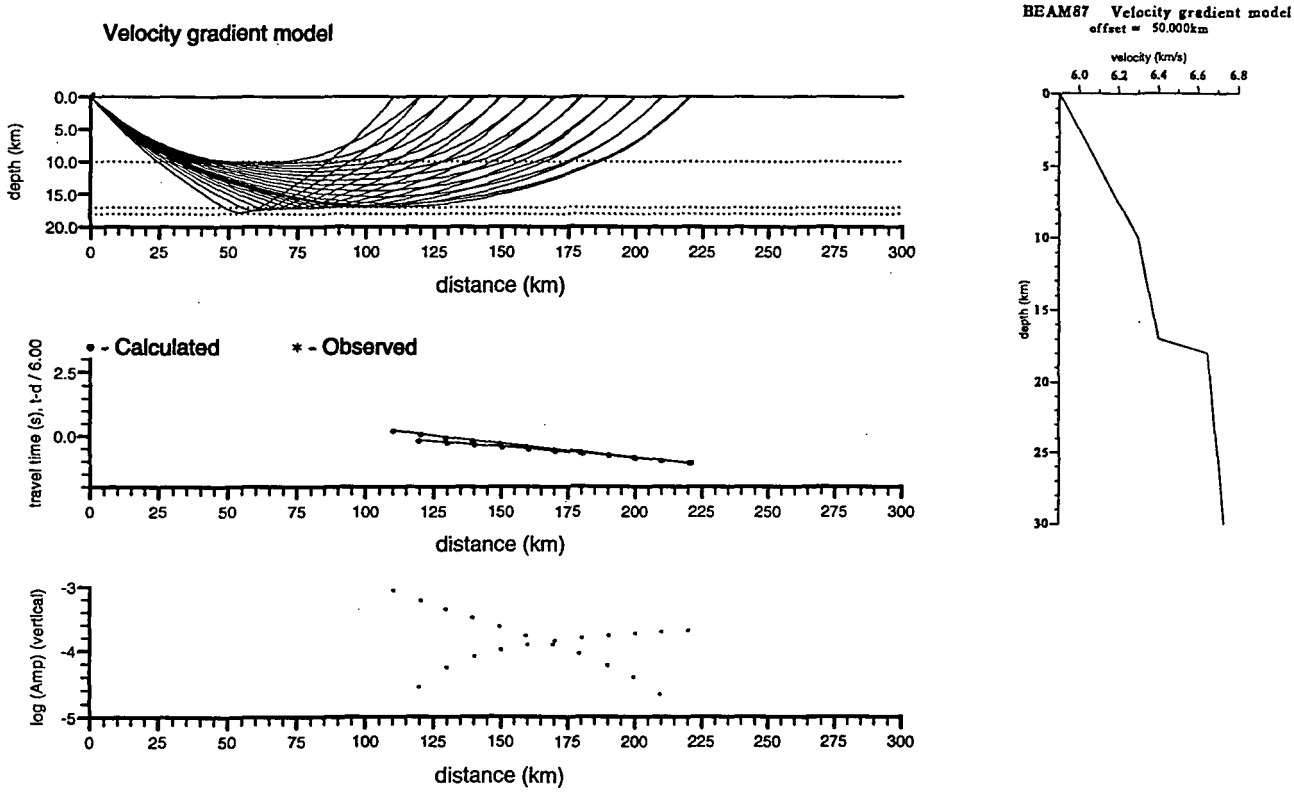
The first problem is that the wide angle model is produced from diving rays and a limited number of reflection events, many of which arise from short segments of a reflecting boundary. As mentioned in section 8.1, the modelling package requires long continuous boundaries. Therefore the reflectors actually giving rise to observed phases are extrapolated across the model and joined up with other such reflecting boundaries. Related to this is the fact that, due to the ray geometry, many reflectors are not imaged at all at wide angle, especially near-surface features at relatively long offsets from any receiver. These reflectors will be seen in the normal incidence data but will be absent from the model and hence from the normal incidence synthetic.

A related problem is the fact that near surface rocks beneath the source are not imaged in wide angle data and therefore not included in the wide angle model. In the real normal incidence section reflection amplitudes will be affected by the transmission qualities of these rocks but in the synthetic section they will not.

The down- and up-going wavefields for each phase in wide angle data experience a different geological column whereas in normal incidence data the up- and down-going wavefields experience the same geological column. The result of wide angle modelling will be to a certain extent an average.

Wide angle arrivals are frequently complex and often have a long wave train. They are generally interpreted as simple impulsive arrivals and modelled as such. The complex nature of the true arrival may be due to





8.2 Ray diagrams and travel time curves comparing reflected arrival with diving ray from a thin, high velocity gradient layer.

scattering from a group of short-length reflectors which will be modelled as a single reflector. In the normal incidence data such a cluster of short reflectors would appear as a patch of high reflectivity rather than a continuous reflection event.

Wide angle data generally contain less high frequency information than normal incidence data due to the longer raypaths involved. The limit of resolution for thin layers is therefore smaller for normal incidence data. Thus layers which are just resolved at normal incidence may appear as a single boundary at wide angle. Conversely, a transition zone between two velocities with thickness intermediate between the dominant, longer wide angle and shorter normal incidence wavelengths will appear as a reflector in wide angle data, where it approximates a velocity step and produces a reflection in the normal incidence synthetic section, but be entirely absent, from, or very weak in, the real normal incidence data.

In a similar vein, it is very difficult to distinguish between a true reflected arrival in wide angle section for which the subcritical part is not apparent, and a diving ray in a thin layer with a high velocity gradient (Fig. 8.2). The former would give rise to a reflected arrival at normal incidence whereas the latter would not.

Reflection amplitude varies with angle of incidence, being greatest near the critical angle. Certain reflectors will therefore appear prominent at wide angle and might be modelled as large velocity discontinuities while others, due to a different incidence angle, may have a much lower amplitude and be overlooked at the interpretation stage. Both reflectors would appear in the observed normal incidence data while only the first would appear in the model and synthetic section.

8.3 Comparison of BABEL models

BABEL Line 1 has also been modelled by Heikkinen and Luosto (1992) using off-line data from recording stations along the Finnish coast (Figs. 4.1, 8.6). Line 6 was modelled by Matthews (1993, Fig. 8.7) and Line 7 by both Bruguier (1992, Fig. 8.8) and Heikkinen and Luosto (1992, Fig. 8.9), all from in-line stations. Fig. 8.10 shows 1D velocity-depth profiles from different parts of each model.

The different models show similar velocity-depth trends, if possible lateral variations are taken into account. However, discontinuities are placed at different depths and the velocity contrasts correspond to these depths so that the models appear somewhat different. If the general velocity-depth trends are to be believed, this would point to a fairly continuous change of velocity with depth broken by short, discontinuous reflecting bodies which do not affect the overall velocity gradient. Since the models are produced from different datasets with different viewpoints, different short length reflectors are imaged causing velocity discontinuities to be placed at different levels in the model.

A noticeable difference between the models is the low velocity zone in Heikkinen and Luosto's Line 1 model (hereafter referred to as Model 1H, Fig. 8.6) which corresponds in depth to the high velocity block in Model A (Fig. 8.3) of this study. Line 6 (Fig. 8.7) has high velocity blocks in the upper crust at a similar depth to those in Model A although they are somewhat thinner. This is in agreement with the gravity anomaly due to the high density of this block which decreases in magnitude towards Line 6. The southern end of Line 1 Model C (Fig. 8.5) also has a high velocity

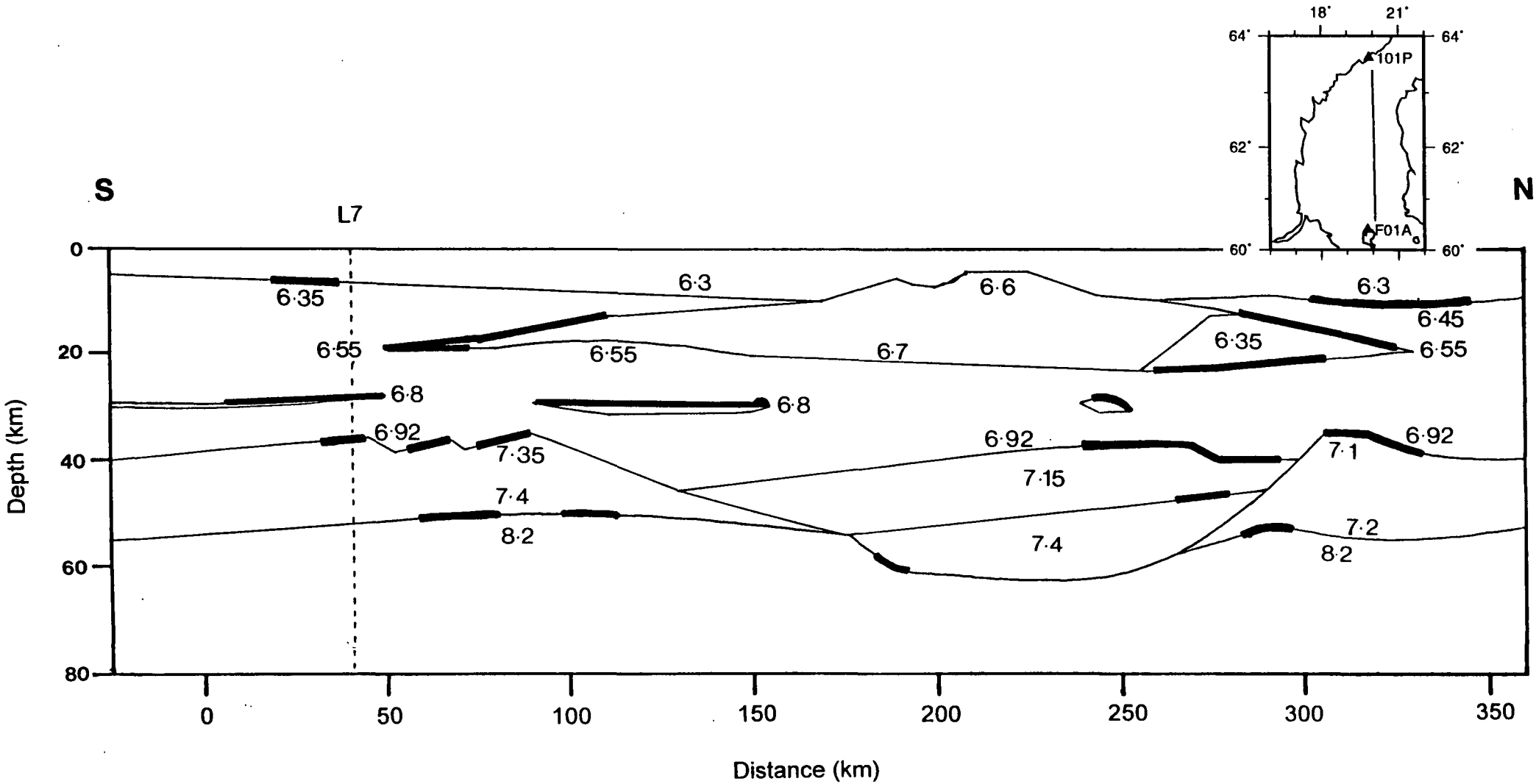


Fig. 8.3 Model A for Line 1 from stations F01A and 101P. Heavy lines indicate segments of boundary imaged in the wide angle data.

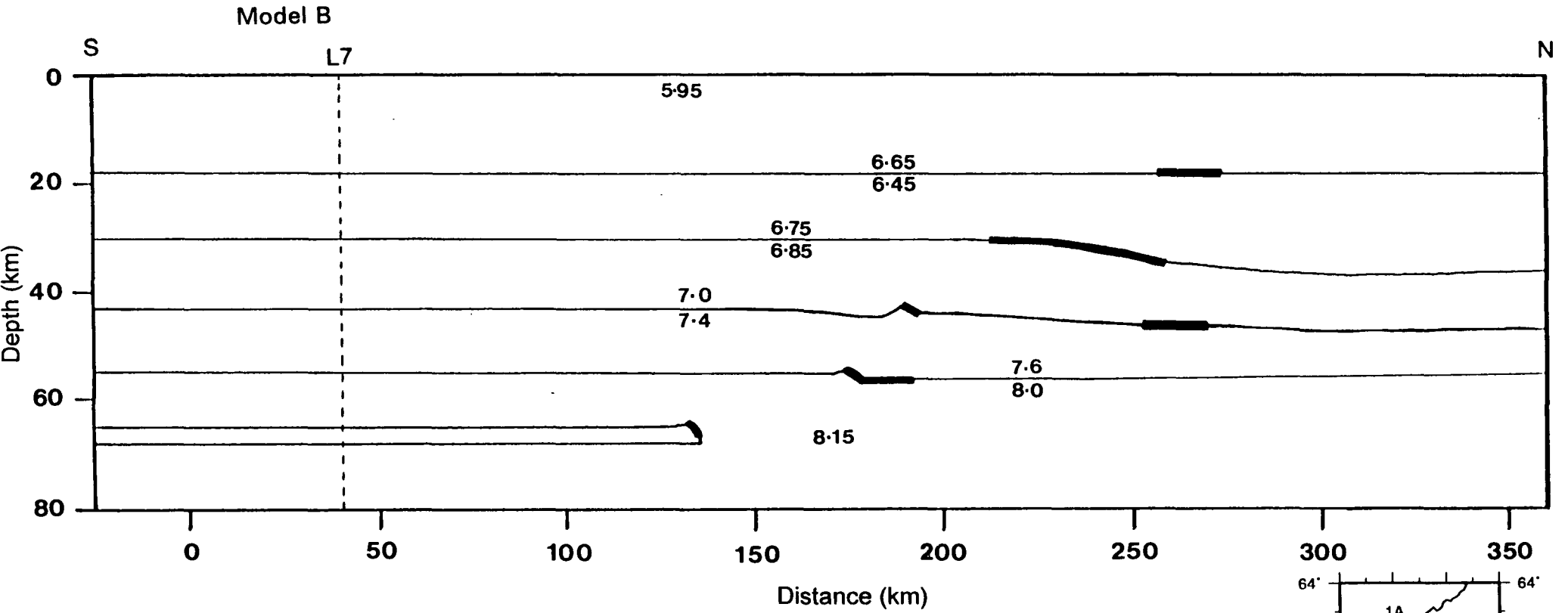
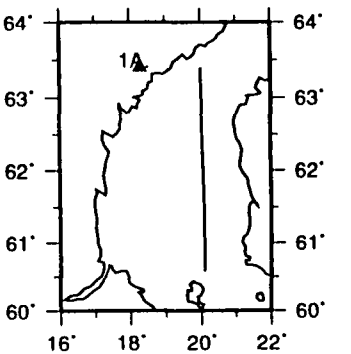


Fig. 8.4 Model B for Line 1 from station 1A.



The highlighted regions indicate segments of the boundaries imaged in the wide angle data.

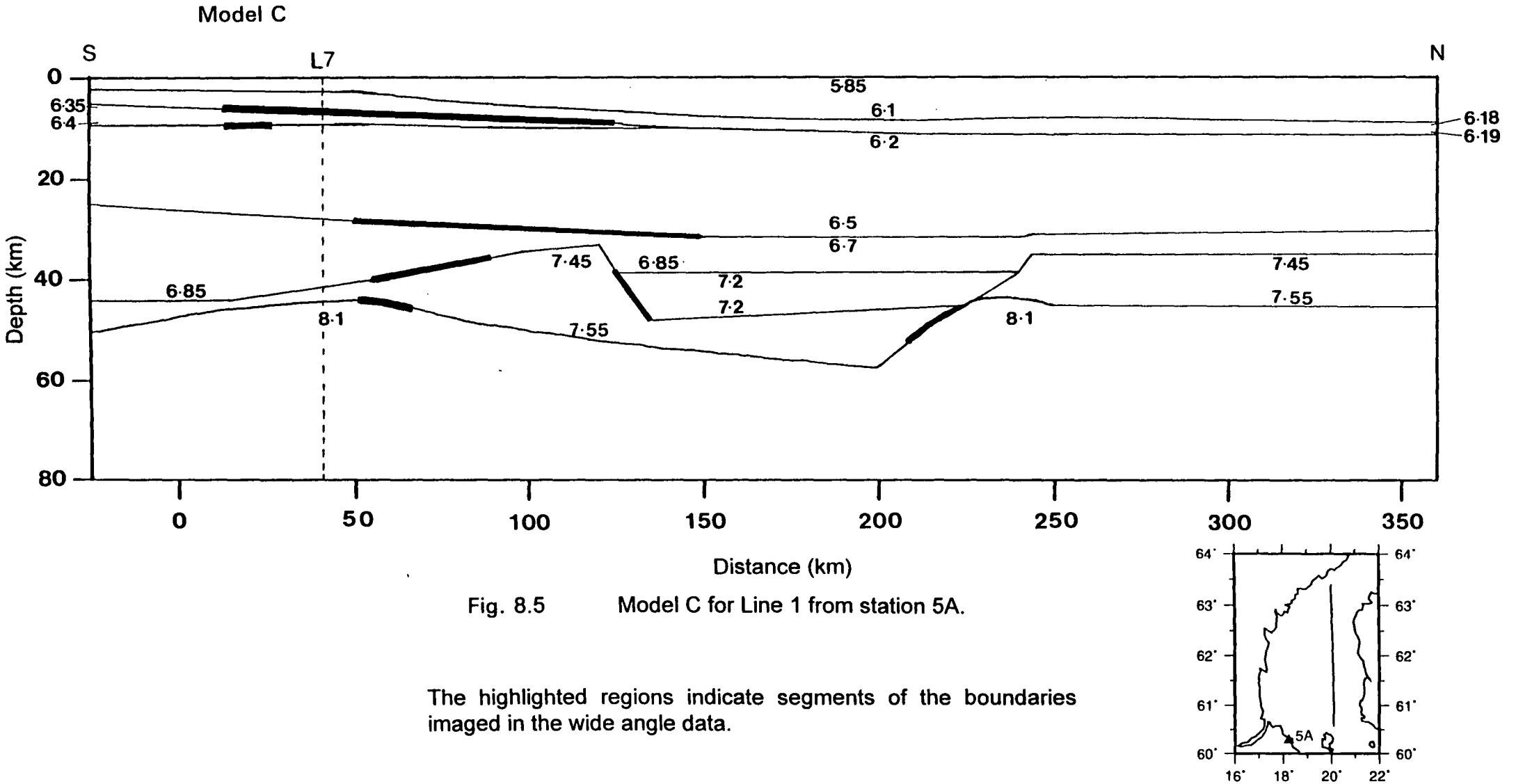
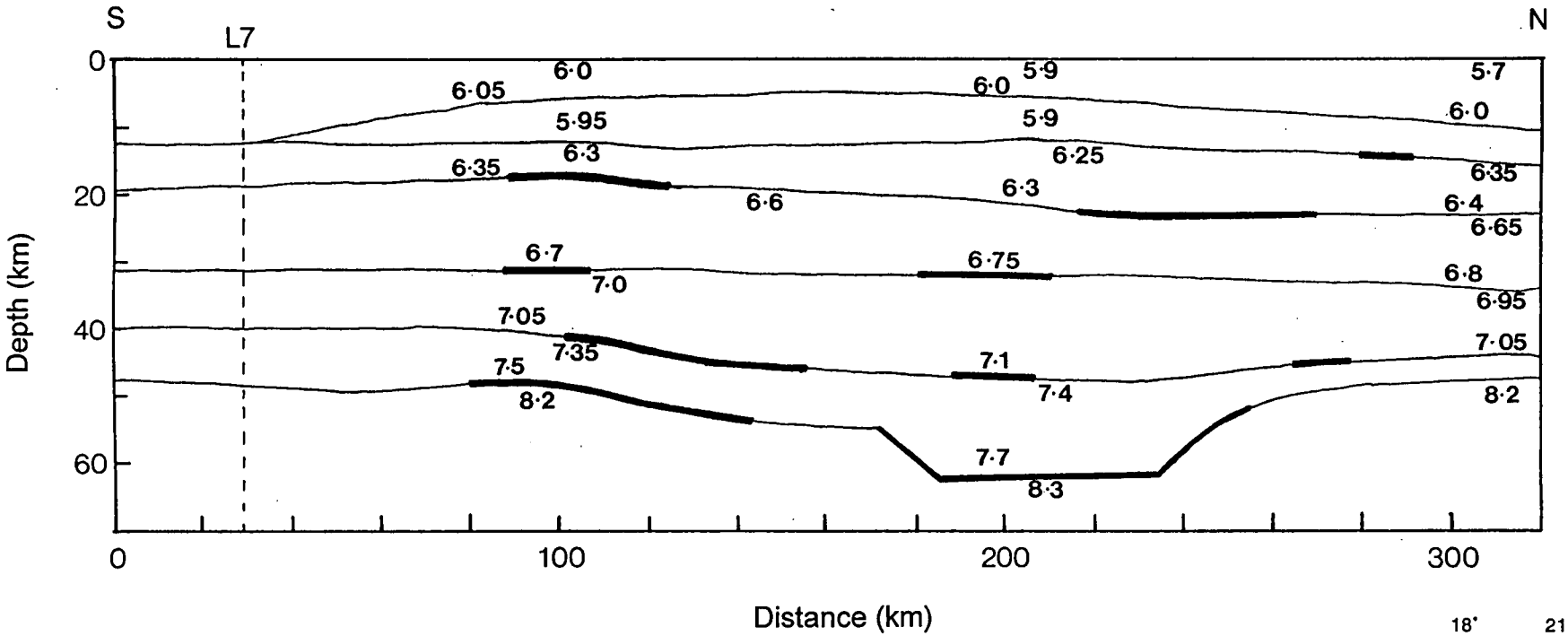


Fig. 8.5 Model C for Line 1 from station 5A.

The highlighted regions indicate segments of the boundaries imaged in the wide angle data.



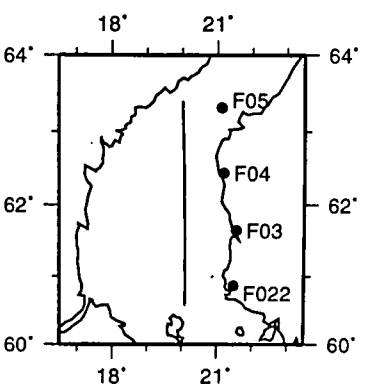


Fig. 8.6 Model for Line 1 from stations in Finland , after Heikkinen and Luosto, 1992.

The highlighted regions indicate segments of the boundaries imaged in the wide angle data.

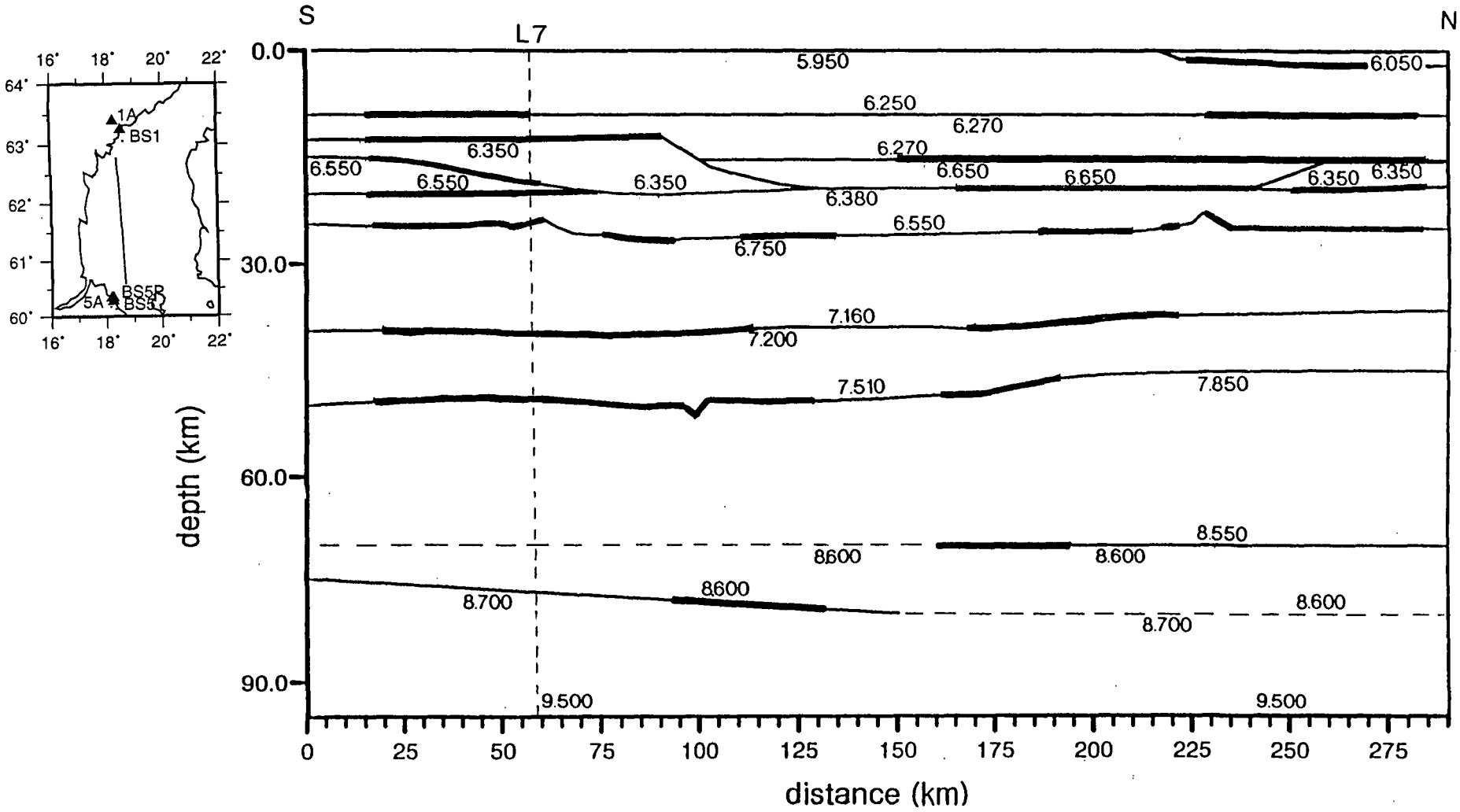


Fig. 8.7 Model for Line 6, after Matthews, 1993.

The highlighted regions indicate segments of the boundaries imaged in the wide angle data.

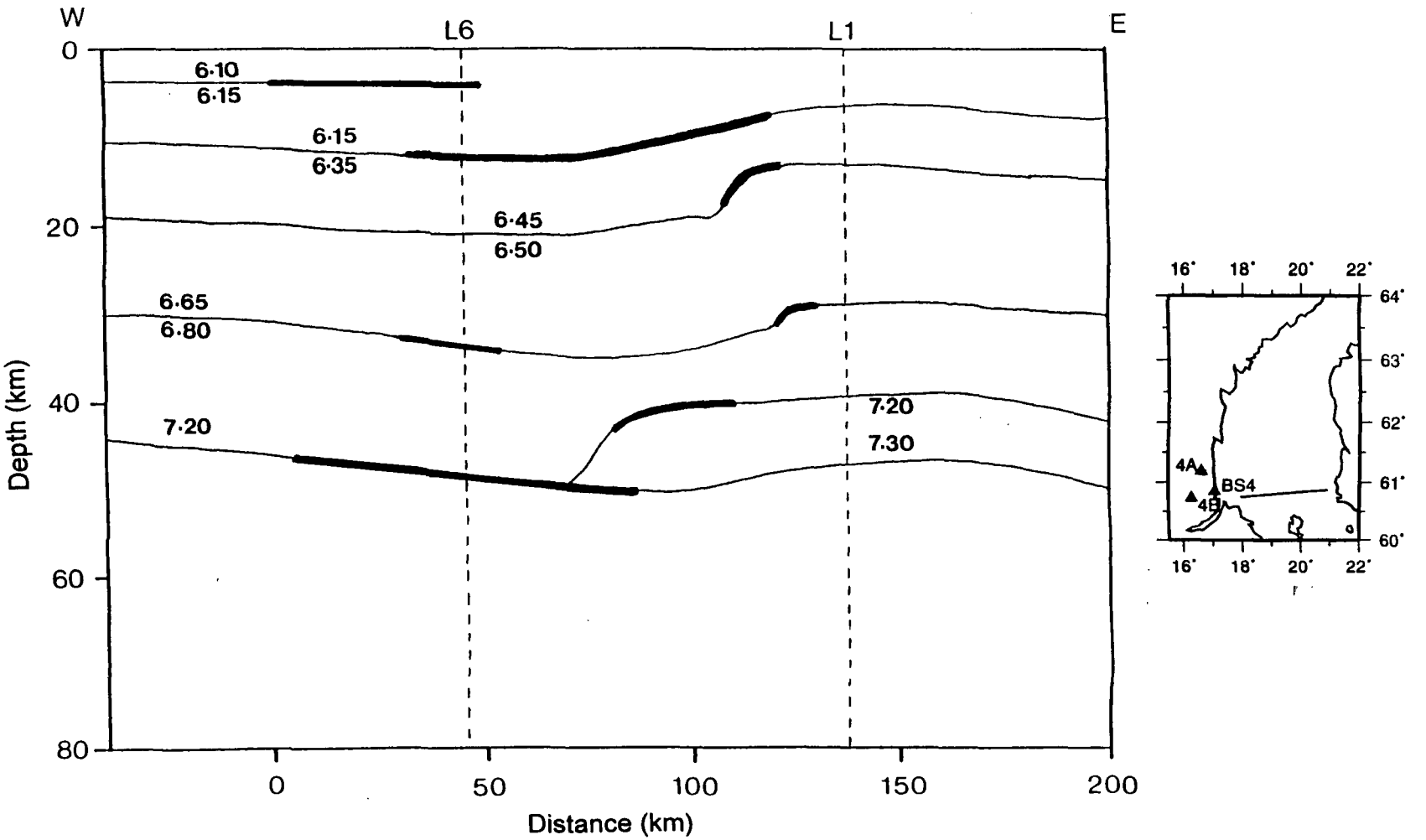


Fig. 8.8 Model for Line 7, after Bruguier, 1992.

The highlighted regions indicate segments of the boundaries imaged in the wide angle data.

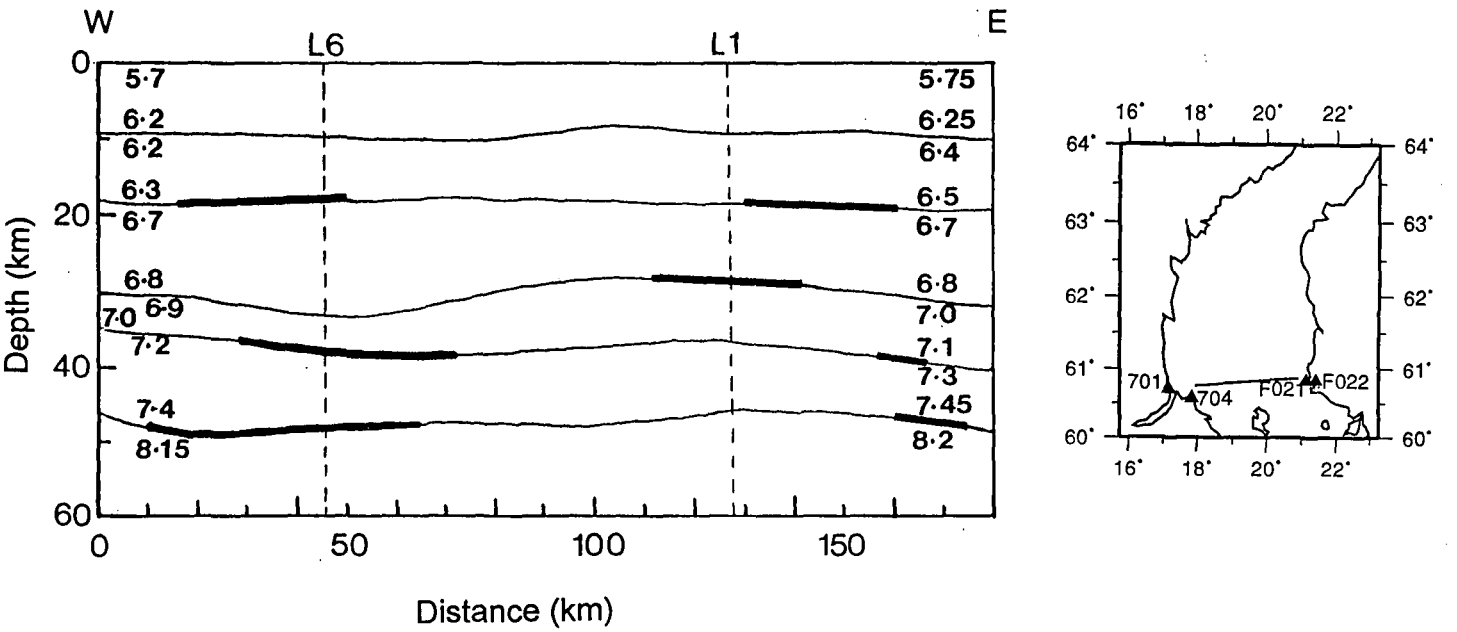


Fig. 8.9 Model for Line 7, after Heikkinen and Luosto, 1992.

The highlighted regions indicate segments of the boundaries imaged in the wide angle data.

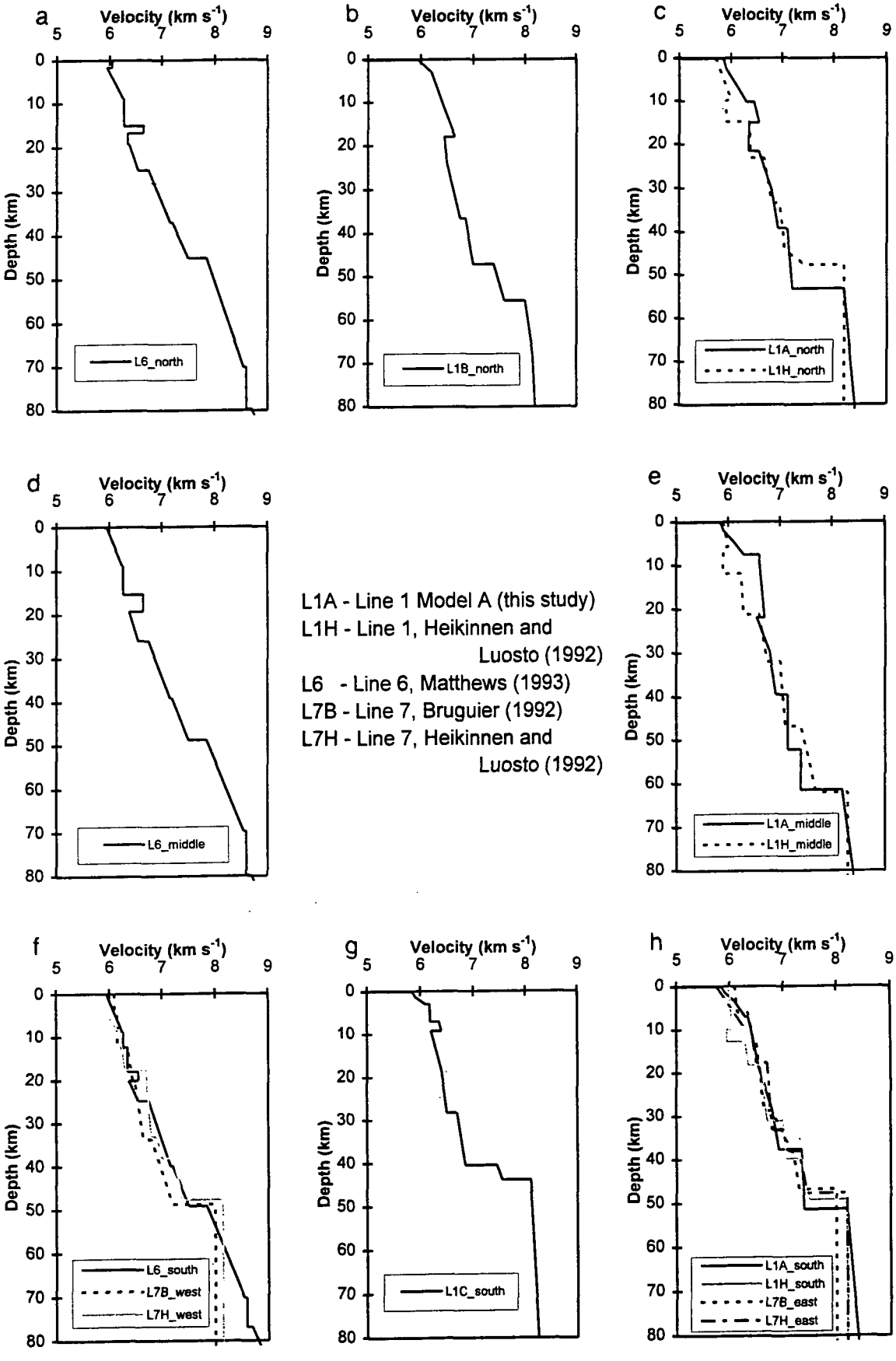


Fig. 8.10 Velocity profiles from BABEL models in the Sea of Bothnia

block which may correspond to that at the south of the Line 6 model though it is nearer to the surface.

In the model for Line 1 station 1A (Model B, Fig. 8.4), although structure in the uppermost crust is not imaged due to the distance of the nearest shot, the upper crustal velocity depth profile is similar to that at the northern end of the Line 6 model. The drop in velocity at 18 km depth corresponds roughly to the bottom of the high velocity zones in both Model A and Line 6.

Both Model A and the Line 6 model exhibit what amount to two diffracting points at a depth of around 30 km, though in Line 6 these have been modelled as irregularities at a velocity boundary rather than isolated bodies. In Model B a dipping reflector is modelled at the equivalent position to the northern diffractor in Model A, but Model C and Heikkinen and Luosto's Model 1H have only continuous reflectors at this depth. It should be noted that only short portions of the reflector are imaged in the Finnish model leading to the consideration that this may alternatively be modelled as an intermittent reflector similar to that in Model A.

The differences in lower crustal velocity structure between Model A and Model 1H may be explained in part by the different datasets used. This means that different 'viewing angles' are employed, and the lateral displacement of the Finnish stations from the line means that the depth points modelled are also displaced to the east of the line in a similar way to those for the off-line stations 1A and 5A (see Chapter 7 section 5), though Heikkinen and Luosto do not specify how they treat their off-line data. They also used a different method of specifying their model so that lateral variations could be incorporated into each layer. However, the depression of the Moho is still apparent in both models at the same position (0 km in

the Finnish model corresponds roughly to 25 km in the reversed model from this study).

The lower crust of the Line 6 model (Fig. 8.7) seems to bear out the suggestion made in Chapter 7 section 5 that the dip in the Moho in Model A shallows out to the west (Fig. 7.33d). Matthews (1993) attempted to model a depression in the Moho or deepest crustal reflector beneath Line 6 but obtained a better fit with upper mantle sub-horizontal reflectors beneath a relatively flat Moho. The Moho lies at around 50 km depth which at the southern end corresponds quite well to Line 1 Model A, though that in Model C is a few km shallower. At the northern end of the Line 6 model the crust is much thinner than that in Model A and Model B. This could be put down to the effects of the Moho depression beneath Line 1 thinning to the west though Matthews (1993) suggests that there is crustal underplating in this region not imaged in the wide angle data. It is interesting to note that the reflecting boundaries in the mid and lower crust of Model B correspond closely in depth with those of the northern end of Line 6. The overall velocity gradient, however, is steeper in the Line 6 model so the Moho along Line 6 is modelled at the same depth as a lower crustal reflector in Model B, ~10 km above the Model B Moho.

Of the Line 1 models, only Model B shows any indication of subcrustal reflected arrivals. This appears in the form of a diffraction as opposed to the horizontal and subhorizontal reflectors modelled for Line 6.

The Line 7 models should provide a way of tying in Lines 1 and 6, intersecting them both at their southern ends. Figs. 8.10 f) and g) show the velocity-depth profiles at the intersections of Line 7 with Lines 6 and 1 respectively. If the models are consistent, the velocity functions at these cross-over points should be identical. Heikkinen and Luosto's model

(Model 7H, Fig. 8.9) is reversed whereas Bruguier's (Model 7B, Fig. 8.8) is unreversed, using data from the western end only. Bruguier's model shows large steps or diffracting points in boundaries at offsets between 80 and 120 km along the model. These features occur at similar depths to laterally limited features in Line 1 Model A. The largest offset is not great enough for reflectors deeper than about 50 km to be imaged but the model suggests that a lower crustal layer exists at the eastern end of the line which is not present in the west. It also implies that the middle crust undergoes some sort of change from west to east, the changeover occurring between the intersections of the two north-south lines. This would explain the differences between Lines 1 and 6 and provides some evidence for the existence of the Baltic-Bothnian megashear proposed by Berthelsen and Marker (1986) in that large scale movement on the shear could give rise to a such a change in crustal structure across it. The evidence, however, is hardly conclusive since the model is unreversed.

The velocity structure of the reversed Line 7 model produced by Heikkinen and Luosto (Fig. 8.9), on the other hand, shows no signs of boundary steps or diffractors but bears greater resemblance to both Line 1 and Line 6. The correlation with Model A is in fact extremely good, and that with the Line 6 model only breaks down in the upper crust. No evidence is seen for the slight crustal thinning between the two lines seen in Model C. This Line 7 model provides a link between the velocity profiles beneath Lines 1 and 6 but it is too far south to explain the disappearance of the Moho depression from west to east.

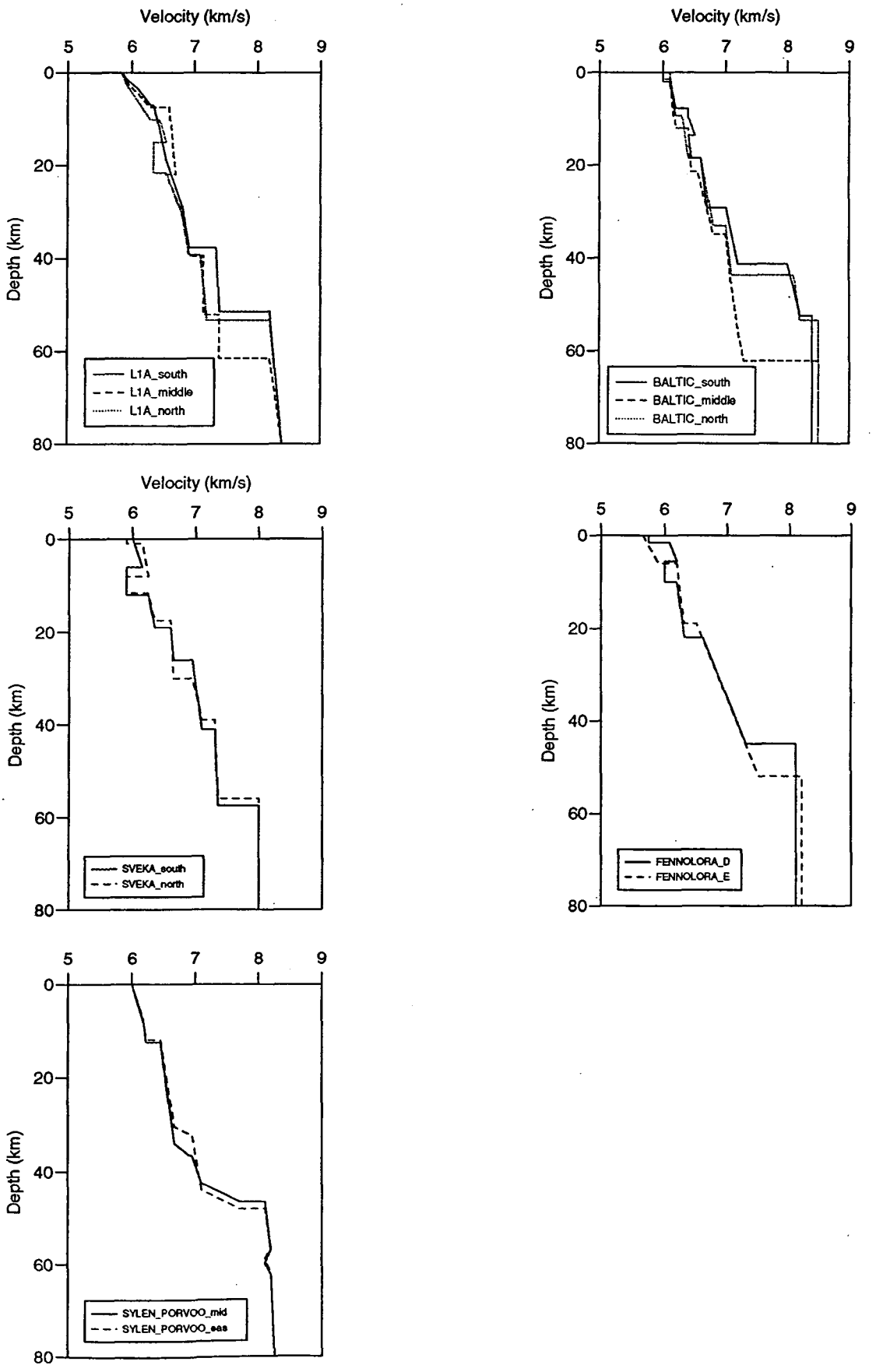
The 2½D magnetic modelling carried out by Pedersen et al. (1992) along Line 7 gives no information about deep structures but indicates a magnetic body in the upper crust at 10 to 15 km depth, between the

intersections with Line 1 and Line 6. It is claimed that a domed reflector can be seen in the normal incidence data, however neither of the wide angle seismic models for Line 7 show any evidence for such a body. According to the magnetic model, the body crosses the intersection point with Line 1. Again little evidence is seen for this in any of the Line 1 seismic models except for Model C which features a thin high velocity layer between 5 and 10 km depth at the southern end, pinching out to the north (Fig. 7.28 Layer 3).

8.4 Comparison with previous projects

In comparison with previous seismic projects (see Chap. 3 and Fig. 8.11), BABEL Line 1 Models A and B have a relatively steep velocity gradient in the upper crust, exceeding 6.55 km s^{-1} at a depth of less than 20 km, and have no distinguishable 'C-boundary'. There is a gradational increase in velocity from ~6.4 km s^{-1} at 10 km depth to 6.8 km s^{-1} at 30 km depth in Model A while Model B undergoes a decrease in velocity at 18 km depth, increasing to 6.75 km s^{-1} at 30 km depth. In the models from previous surveys the velocity only exceeds 6.6 km s^{-1} below the C-boundary at approximately 20 km depth. The C-boundary would appear to occur somewhat deeper at around 30 km depth in Model C and the Line 6 model. Both the BALTIC and FENNOLORA models (Figs. 3.3a and 3.3b respectively) exhibit high velocity zones in the upper crust at their southern ends which may correlate with similar zones in the Line 1 and Line 6 models, while in the SVEKA model (Fig. 3.3b) there is a low velocity zone at ~10 km depth.

Fig. 8.11 Comparison of 1D velocity profiles from BABEL Line 1 with previous seismic projects in the Baltic Shield (See Chapter 3 for references).



Between 20 and 40 km depth, all the models are fairly similar in terms of velocities and overall gradients, though the depths of discontinuities vary considerably from model to model.

The boundary at ~40 km depth in all three Line 1 models is taken as representing the top of the lower crust. Below this significant velocity discontinuity, the velocity exceeds 7.0 km s^{-1} and at the northern and central parts of the line there is a distinct increase in the normal incidence reflectivity (Fig. 8.1). Pavlenkova (1979) and Nelson (1991) cite both of these as characteristics of Precambrian lower crust.

This boundary corresponds well to the top of the lower crust in the SVEKA model (Luosto, 1990) and western end of the Sylen Porvoo model (Luosto, 1986) though in other models such as BALTIC and BLUE ROAD, the Moho lies at this depth. The depth of the Moho depression in the middle of Model A is similar to that in the BALTIC model though the Moho depth to either side correlates better with an upper mantle reflector in the BALTIC model. This poses the question of whether the Moho has been correctly identified in the BALTIC model (that in BABEL Line 1 Model A being corroborated by the base of reflectivity in the normal incidence data) or whether the crust is actually thinner. The crustal thickness beneath shotpoint E of the FENNOLORA profile is also in good agreement with that at either end of Model A and Model B.

The BALTIC, SVEKA and FENNOLORA profiles all have dips in the Moho reaching down to around 60 km depth. That in the BALTIC model bears the closest resemblance to Line 1 Model A. Also present in the BALTIC model are laterally limited reflectors at 30 km depth similar to those modelled in Model A. The SVEKA model contains boundary irregularities at 30 km depth which may be comparable to diffracting

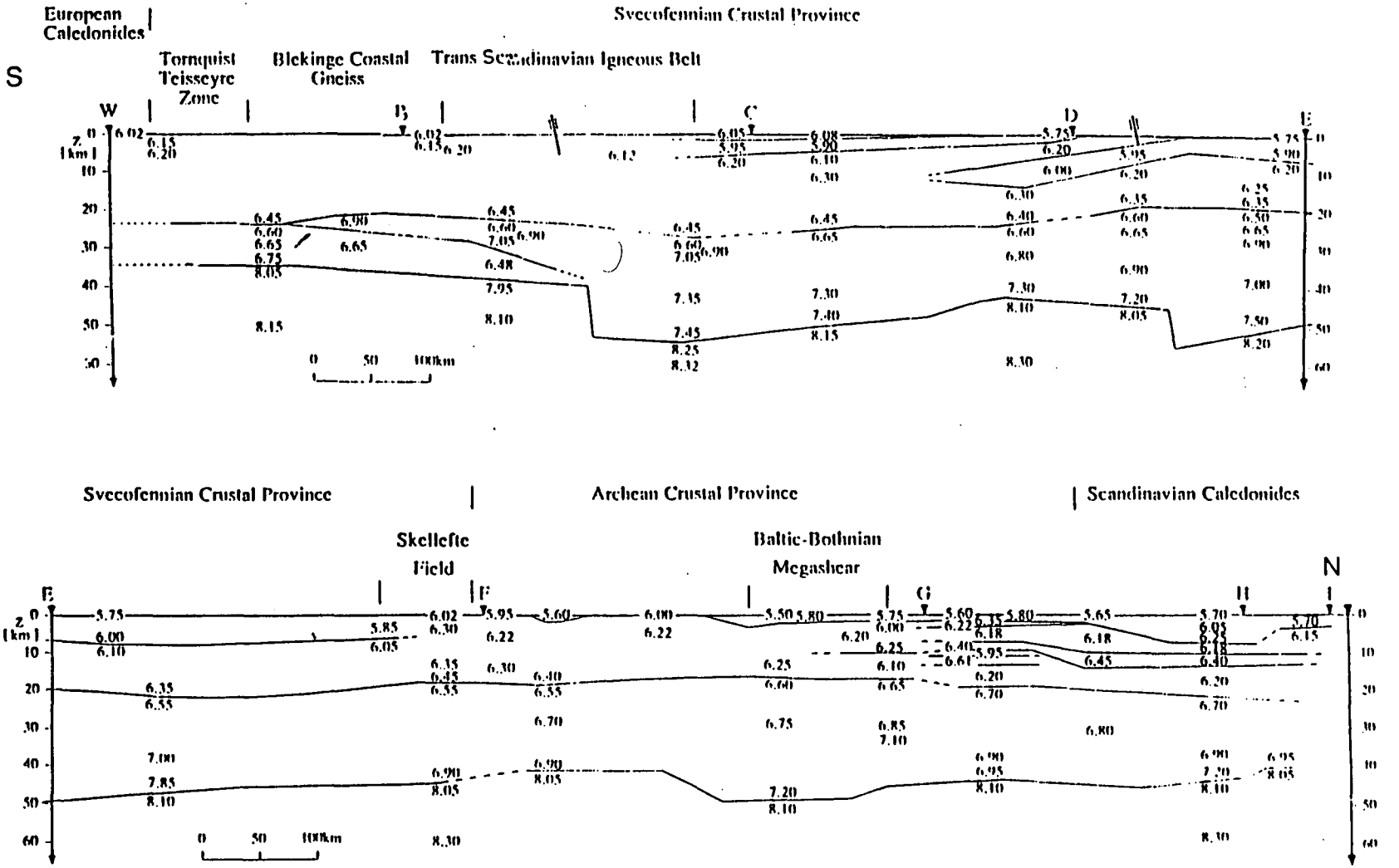


Fig. 8.12 Velocity model of the whole FENNOLORA profile (Guggisberg et al, 1991).

features in Model A, and another at 40 km comparable to the one seen in Model B. That two of these profiles run directly across the Ladoga-Bothnian Bay shear zone lends weight to the possibility of there being a similar megashear beneath the Bothnian Sea (see section 8.3). Fig. 8.12 shows the model of the whole FENNOLORA line with surface geology indicated (Guggisberg et al, 1991). The three Moho depressions here can also be related to surface faulting. On the other hand, this does not explain there being a Moho depression beneath the FENNOLORA line, and not one beneath BABEL Line 6. It may be that such Moho irregularities are common within the Precambrian Baltic Shield, or the answer may lie in Matthew's proposed crustal underplating at the northern end of Line 6 being equivalent to the Moho depressions modelled in Line 1 and FENNOLORA.

No upper crustal structures similar to that in Model A have been modelled above the Moho depressions in the BALTIC or FENNOLORA lines while a high velocity block has been proposed in the upper 10 km of the SVEKA line. However, the spatial resolution of the seismic data along these profiles was much poorer than for the BABEL lines so the same complexity of arrivals will not have been modelled, and without integrated gravity modelling it is difficult to say whether the combination of a high density upper crustal block with a Moho depression is unique to Line 1 or typical of the region.

8.5 Gravity modelling

Gravity modelling based on the seismic model achieved a good fit to the gravity data in two dimensions (Figs. 7.16 and 8.13a). The main, long wavelength, negative anomaly (80 - 350 km in Fig. 7.16) was modelled by

a Moho depression and a lateral change in lower crustal density. Superimposed on this is a shorter wavelength positive anomaly which was modelled by a high density body in the upper crust.

If the reference density (0 kg m^{-3} in the model) is taken to be 2880 kg m^{-3} then the top layer has an average density of 2690 kg m^{-3}, the high density body 2940 kg m^{-3}, the lower crust 3110 - 3160 kg m^{-3} and the mantle 3300 kg m^{-3} (shown in brackets in Fig. 8.13a). These are reasonable values for this type of crust and compare well with models from other Precambrian regions (Fig. 8.14, Goleby et al 1990, Braile 1989). Fig. 8.15 shows a plot of velocity against density for the Line 1 model compared with data from several other Precambrian regions and the Nafe-Drake curve.

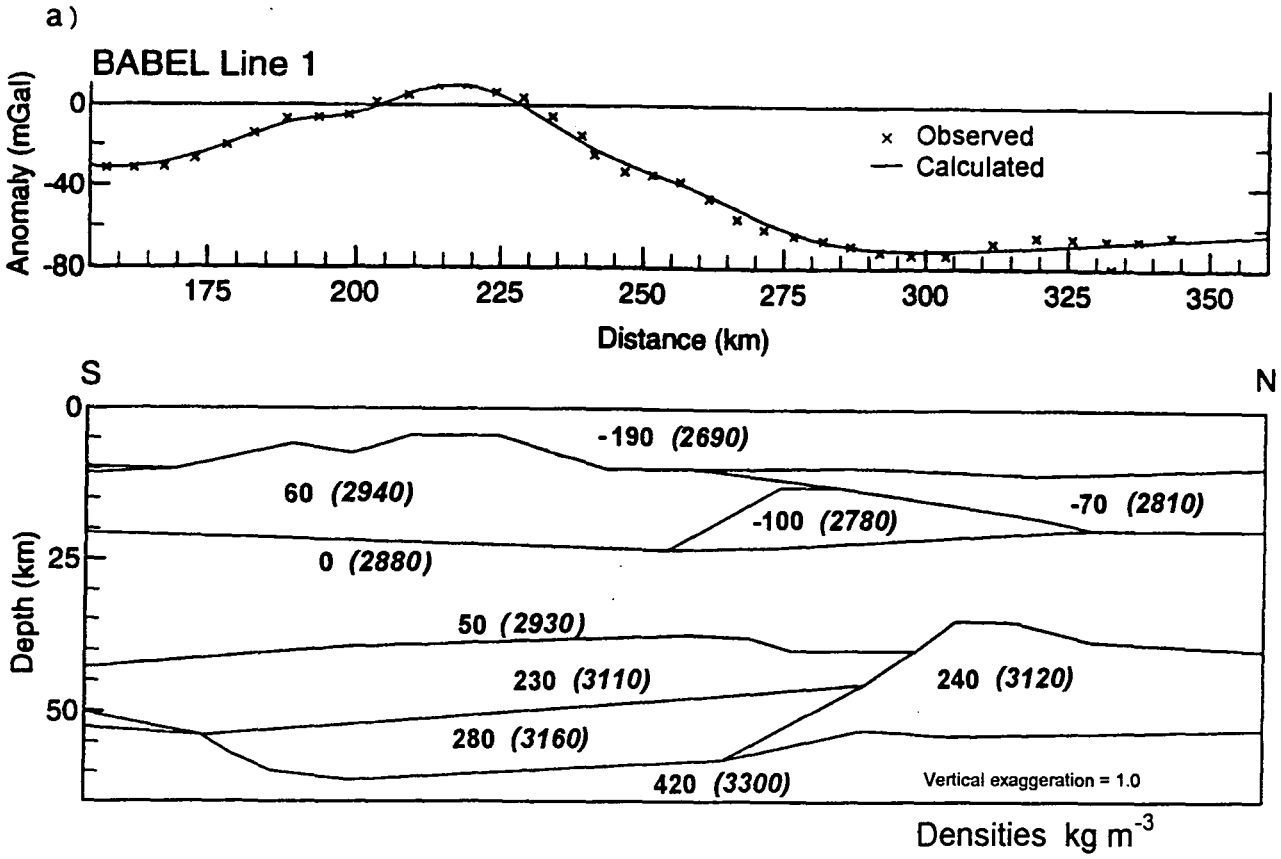
Pedersen et al. (1992) also carried out 2½D gravity modelling along Line 1 using a smoothed gravity dataset. Fig. 8.13b Shows Pedersen's model in comparison with the equivalent part of the model from this study. Pedersen's density contrasts are relative to other values at the same depth only and so the modelled values are not directly comparable. However, densities using the same constant reference value (2880 kg m^{-3}) are shown in brackets to give a rough comparison with Fig. 8.13a. The gravity high is modelled by a 100 km long, 30 km thick body with a density contrast of +120 kg m^{-3}, in a similar position to the high density body in the model from this study. Pedersen interprets this as a dioritic pluton, citing outcrops in nearby Uppland, Sweden, of density 2810 kg m^{-3} as supporting evidence.

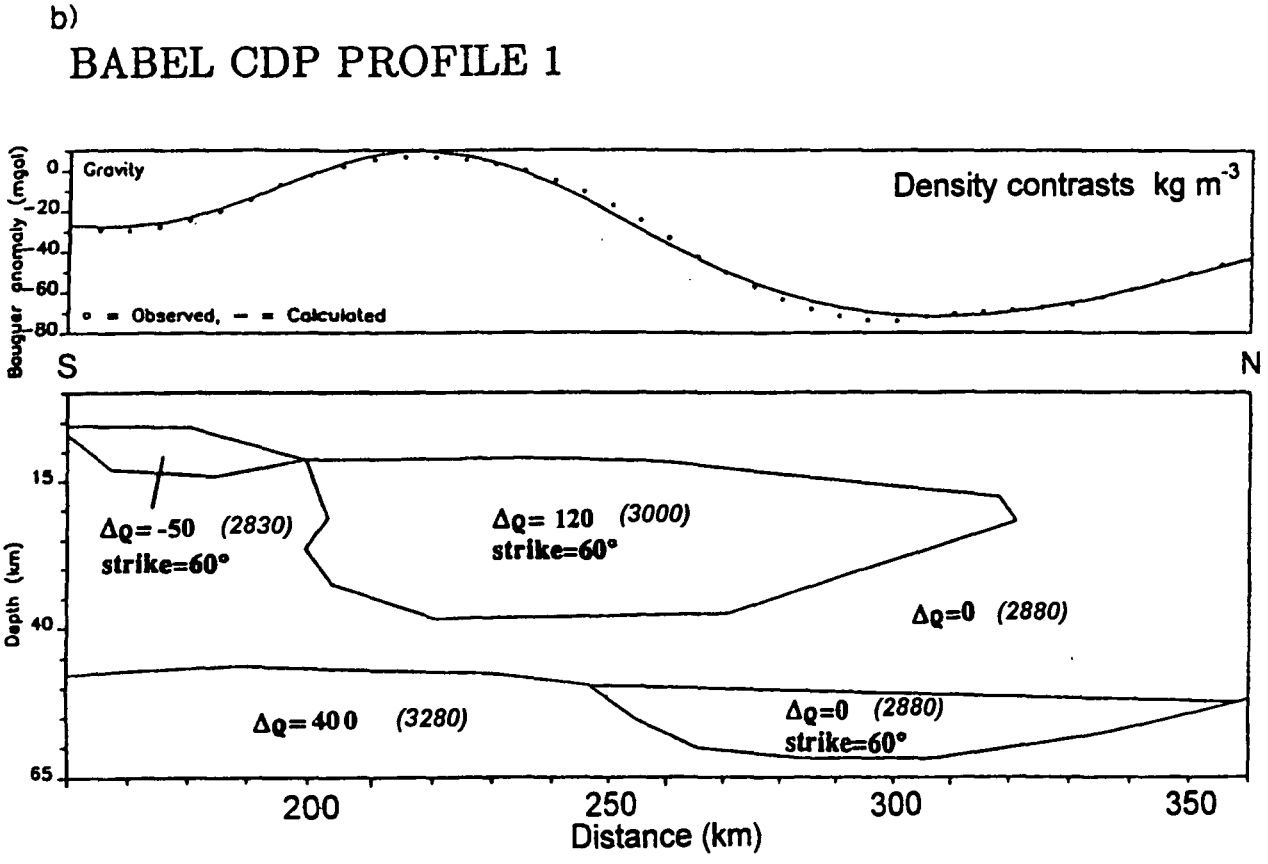
Pedersen's model also indicates a region of crustal thickening to account for the negative anomaly. However this is 60 km north of the Moho depression in the seismic/gravity model from this study and has no supporting evidence from either wide angle or normal incidence seismic

data. Neither is there any evidence for the small, near surface, low density block towards the southern end of Pedersen's model. Clearly, the integrated seismic and gravity model from this study, despite being only 2D, is better constrained than Pedersen et al's 2½D gravity model.

Taking the third dimension into consideration, one must refer back to the gravity anomaly contour map (Fig. 3.5). The extent of the high density near surface body causing the positive anomaly can be inferred from this map (Fig. 8.16b). From this information, the possible location and extent of the Moho depression may be deduced by estimating the gravity anomaly contours if the high density body were absent. To do this, the 2D gravity anomaly for Model A with the high density block removed was calculated (Fig. 8.16a). This shows that the small gravity minimum to the southeast of the positive anomaly is part of the underlying negative anomaly due to the Moho depression and can be correlated with the southern extent of the observed negative anomaly to the west of the positive anomaly. The result is a wide negative anomaly with a general E-W trend, extending onshore into Sweden and dying out to the west, rather than the NE-SW trend apparent from the actual anomaly map.

The upper crustal high density block appears from the gravity map to be more limited in its extent than the Moho depression. It may be that it is only the thickest part of the body, where its top nears the surface, which is so limited (compare Figs. 7.13 and 7.16). The thinner extremes probably extend much further, tying in with the upper crustal high velocity zone in the central part of Matthews' Line 6 model.





8.13 **Comparison of BABEL Line 1 gravity models. Actual densities, given reference value of 2880 kg m^{-3} are shown in brackets.**

a) 2D gravity model along BABEL Line 1 from this study.

b) 2½D gravity model along BABEL Line 1 (Pedersen et al, 1992).

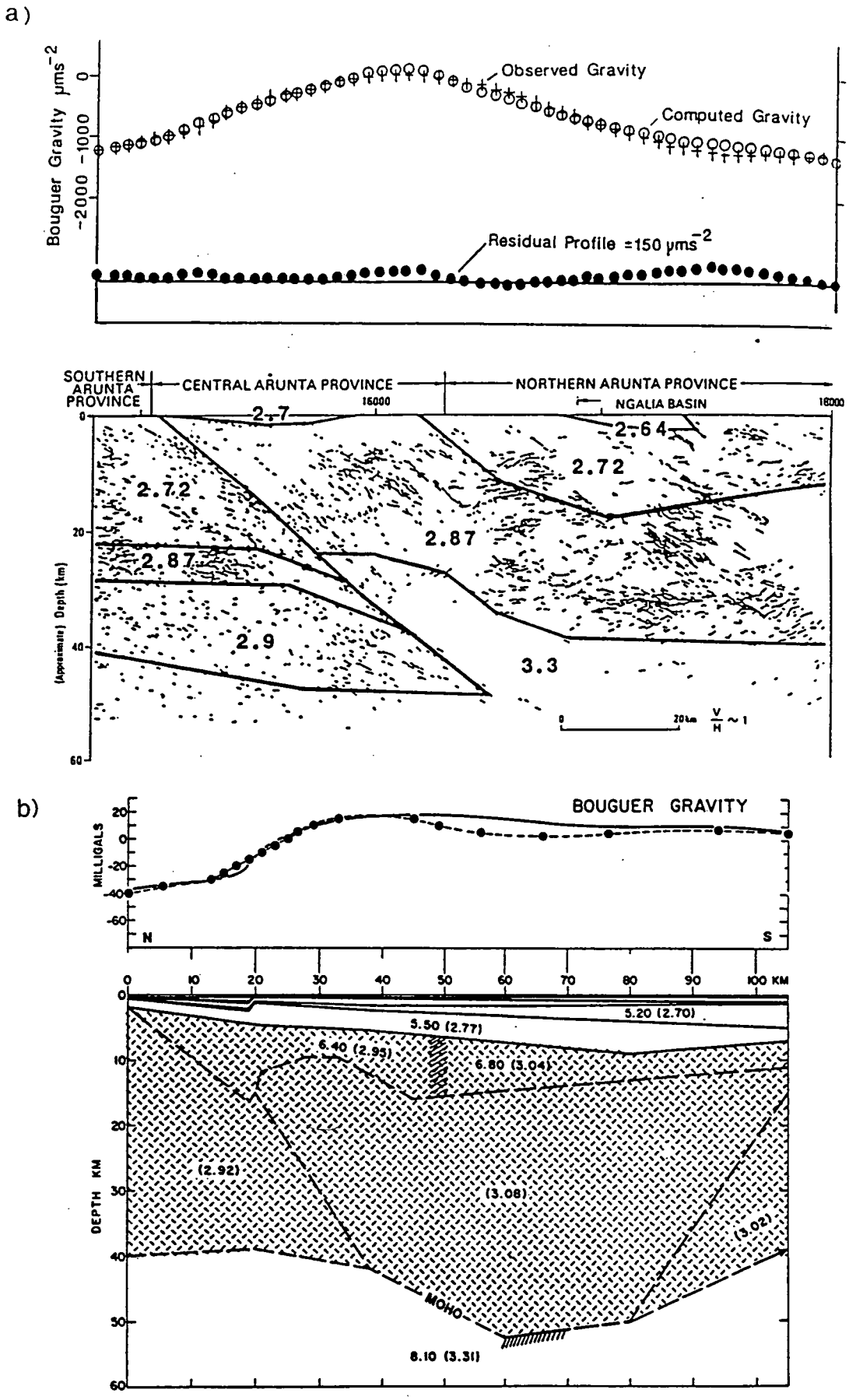


Fig. 8.14 Gravity models from other Precambian regions
a) Proterozoic Arunta Block, Central Australia (Goleby et al, 1990)
b) Western Lake Superior, North America (Braile, 1989).

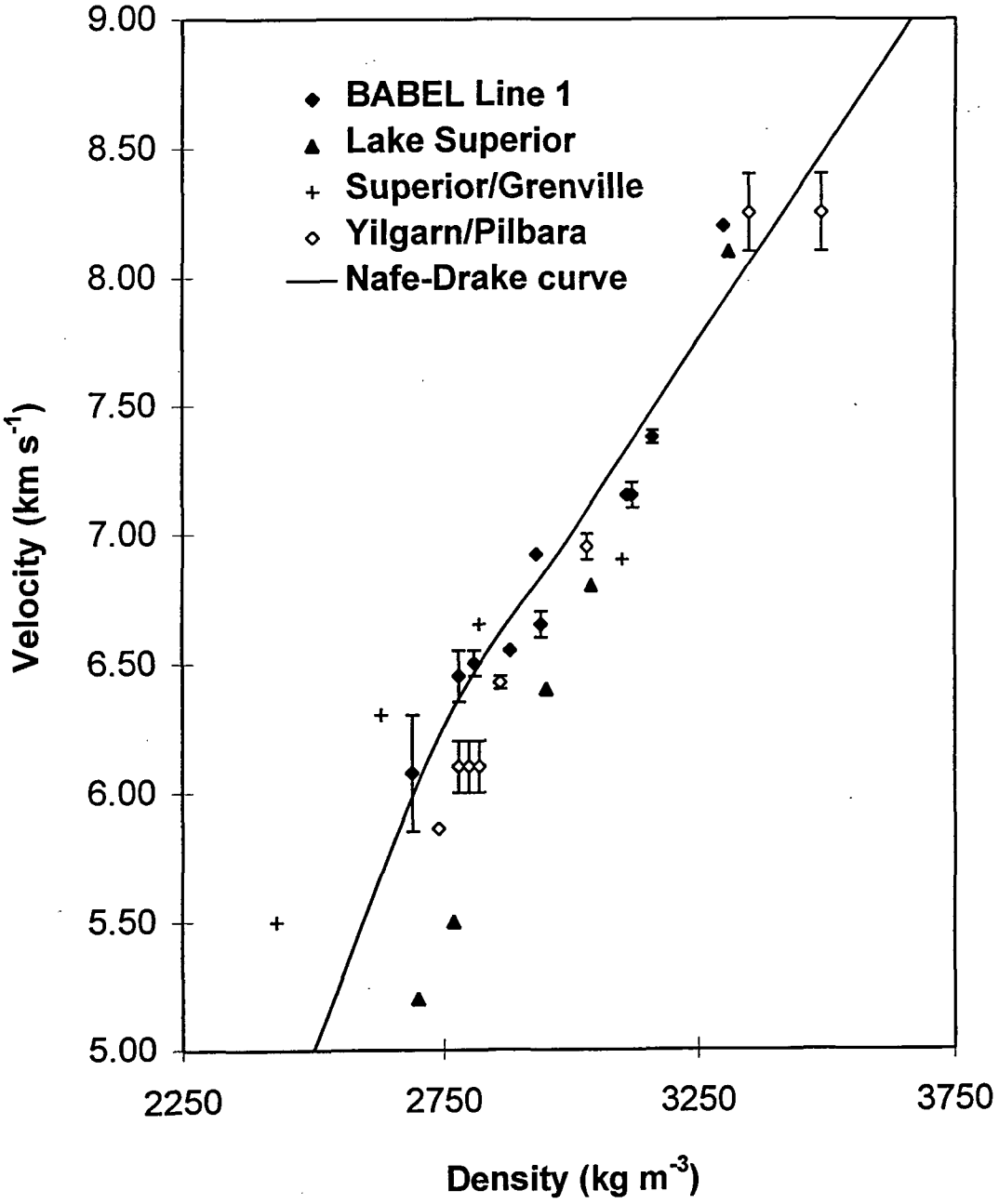


Fig. 8.15 Velocity vs. density for BABEL Line 1 and other Precambrian regions compared with the Nafe-Drake curve.
Lake Superior, N. America (Braile, 1989)
Superior/Grenville Provinces, N. America (Berry and Fuchs, 1973)
Yilgarn and Pilbara cratons, SW Australia (Wellman, 1988)

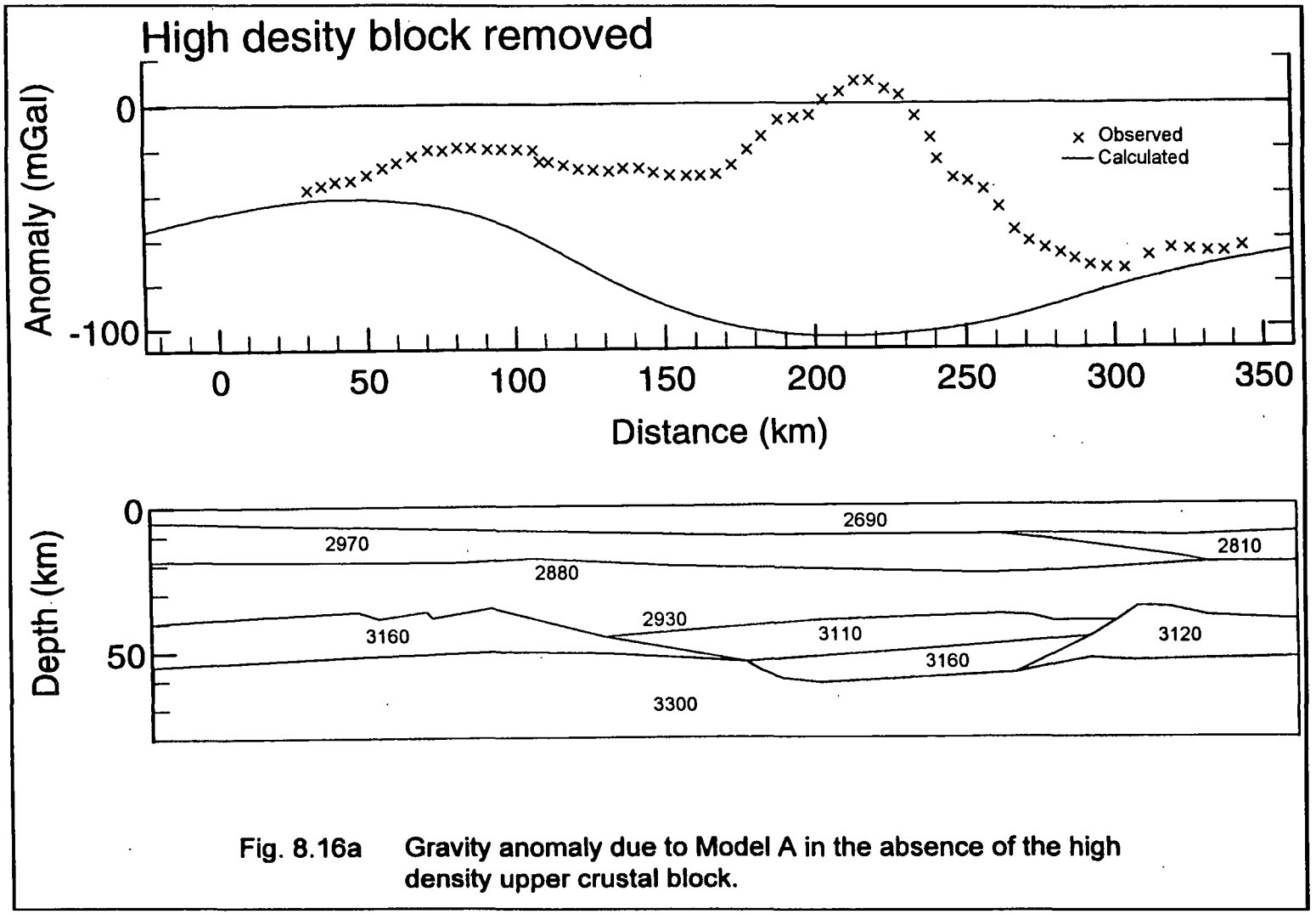


Fig. 8.16a Gravity anomaly due to Model A in the absence of the high density upper crustal block.

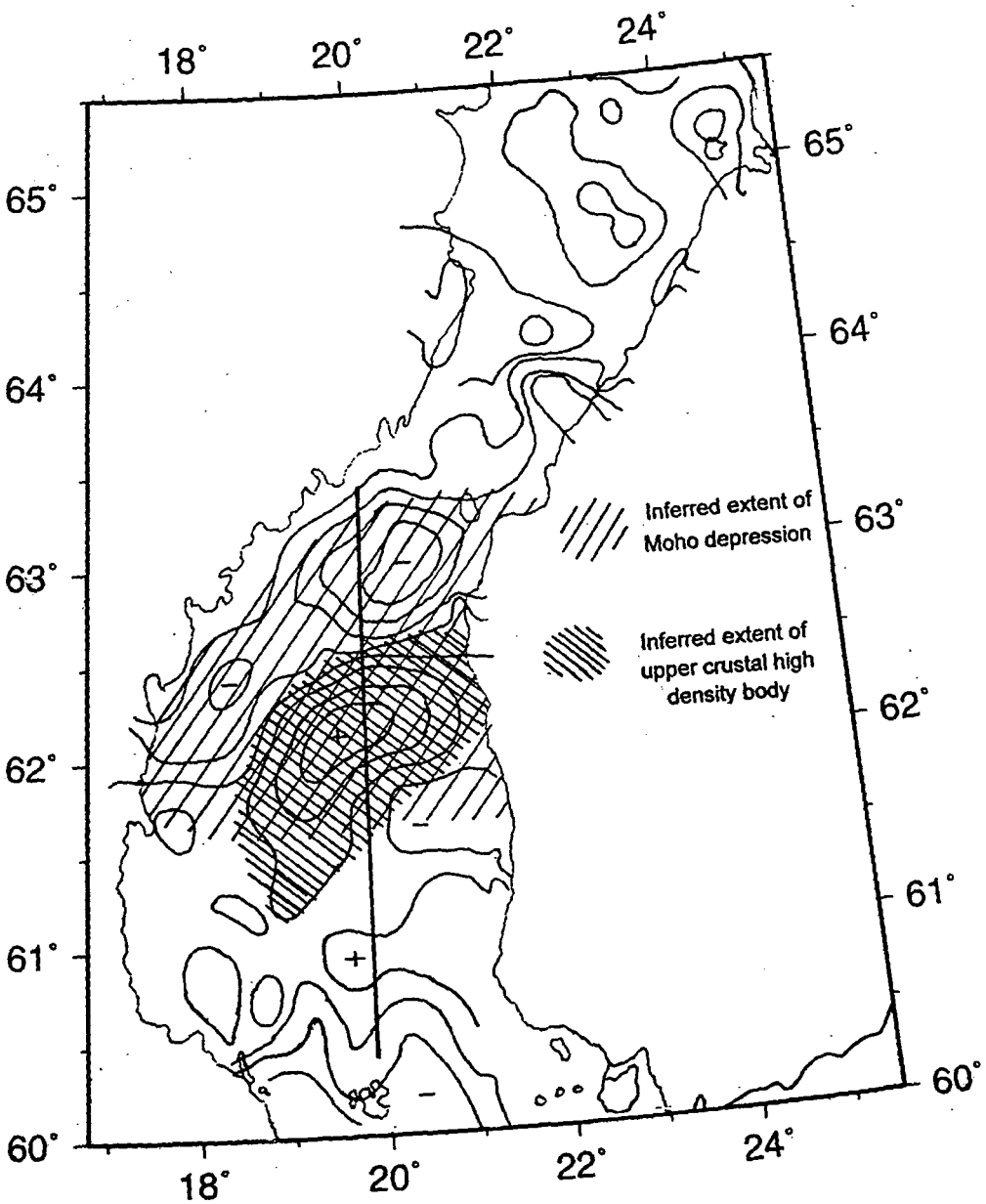


Fig. 8.16b Bouguer anomaly contour map of the Sea of Bothnia with inferred extents of the upper crustal high density body and the Moho depression.

8.6 Geological interpretation

In the upper crust at the northern end of Line 1, the two bright reflectors seen in the normal incidence data have been interpreted as dolerite sills, as has been explained in Chapter 6. These were not imaged directly by the wide angle data, being too near the surface and not close enough to the receiver. For this reason they were not included in the model. They may, however, be responsible for some of the multiples seen in the data at the northern end of the line.

The source of mid-crustal diffractions at around 30 km depth is not known but the normal incidence data would suggest that diffracting or scattering bodies are not uncommon in this section of shield crust, especially towards the southern end of the line. Indeed, two short reflectors are seen in the normal incidence data quite close to the modelled diffracting points (Fig. 8.1). The question should perhaps be why do not more such features appear in the model?

The most significant part of the model seems to be the central structure. The general shape implies either a large, rounded structure, possibly a large, crustal scale igneous intrusion, or two related structures : one in the upper, and one in the lower crust. A large Episode 1 granitoid intrusion (Andersson's nomenclature, 1991) of comparable lateral dimension is seen immediately to the east, on shore in Finland (Fig. 2.3). Gravity modelling, however, indicates a dense intrusive body in the upper crust and also corroborates the thickened lower crust (i.e. Moho depression) and lateral change in lower crustal seismic velocity.

A large mafic intrusion in the upper crust is not implausible. Such a body has also been modelled by Pedersen et al. (1992) and interpreted as

a large dioritic pluton (section 8.5, Fig. 8.13b). It is possible that the body extends deeper into the middle crust where the density and seismic velocity of the surrounding rocks will be more similar. Evidence in favour of this is the transparent region of the normal incidence section in this region. Such transparent zones have previously been associated with plutons in the upper crust which provide a 'window' to the lower crust (Matthews, 1987). Against this view is the presence of reflectors in the normal incidence data which correlate well with the base of the modelled body (see Fig. 8.1).

The presence and cause of the Moho depression is more controversial. The most obvious questions arising are how was it formed ? Why is it still there when one would expect it to relax and flatten over geological time? Indeed, could it merely be an effect of complex velocity variations above?

The presence of the depression as a true geological feature is supported by the reflected wide angle arrivals from its sides (Models A and C), Heikinnen and Luosto's model from data recorded at different locations, and the gravity model. Moho depressions have been modelled elsewhere in the Baltic Shield : beneath the BALTIC and SVEKA lines, three sections of the FENNOLORA line and perhaps also the Blue Road Line. It may be that the same explanation was used for similar data in each successive survey because it worked before, but no alternative models have been published.

Kusznir and Matthews (1988) showed that given an offset on the Moho, different wavelengths would relax at different rates, decaying by ductile flow. Intermediate wavelengths (around 10 km) decay more rapidly than longer and shorter wavelengths, effectively notch filtering the Moho

topography and giving rise to the remnant structure shown in Fig. 8.17, hereafter referred to as the Kusznir-Matthews structure.

This type of structure has been used to model crustal faults which cut the Moho (West 1990). Such a structure is not clearly seen at normal incidence so no obvious Moho offset is observed in the data, though it may be seen in the wide angle data. The Moho depression seen in BABEL Line 1 may be related to the Kusznir-Matthews structure, or possibly two such structures side by side.

To derive a possible explanation for the origin of the Moho topography, one must consider all the available information : locations of seismically modelled depressions, gravity anomalies, conductivity models, the surface geology, and tectonic and fault maps. In Fig. 8.18, all this information is portrayed on the same map.

As explained in section 8.5.2, it is postulated that the gravity low due to the Moho depression beneath Line 1, in the absence of the high density upper crustal body, would be much wider and extend onshore into Sweden, coinciding with the Moho depression beneath shotpoint E of the FENNOLORA profile. A similar gravity low is seen at the Moho depression beneath FENNOLORA shotpoint C in Southern Sweden. To the east, the gravity low appears to coincide with the boundaries of the Central Svecofennian tectonic unit, the edges of the Central Finland Granite complex, strike-slip faults and the Tampere Schist Belt conductivity anomaly. Unfortunately, gravity data for Finland itself was not available.

The Tampere Schist Belt anomaly and associated electromagnetic anomalies form a 500 km long zone running E-W from the Finnish-Russian border to the Bothnian sea (Pajunpää, 1987). The conducting bodies have been interpreted as the result of the collision of crustal terranes on the

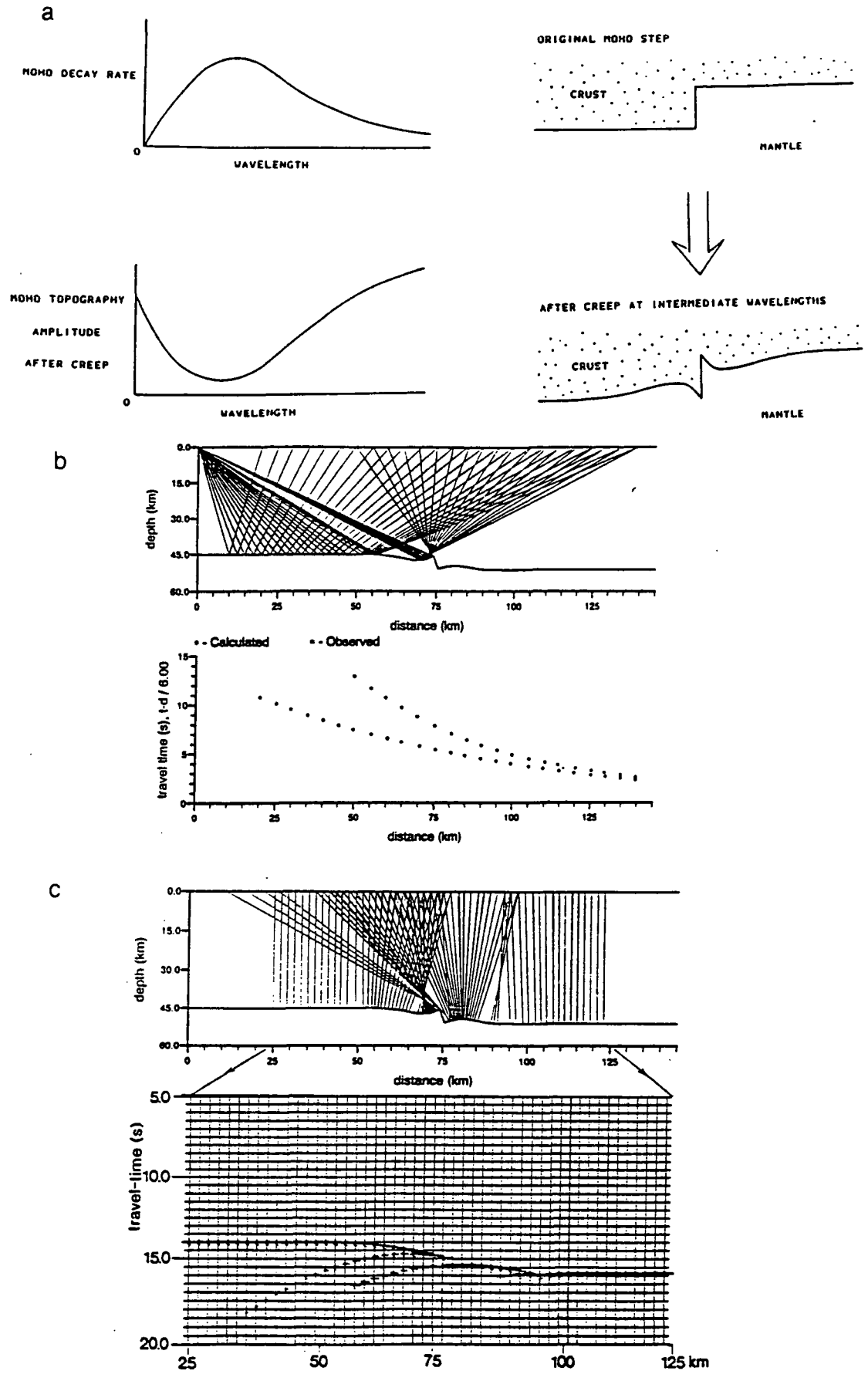


Fig. 8.17 a) Decay of a Moho step over geological time (Kusznir and Matthews, 1988).
b) Effect of the Kusznir-Matthews remnant structure on the wide angle travel time curve.
c) The same structure imaged at normal incidence.

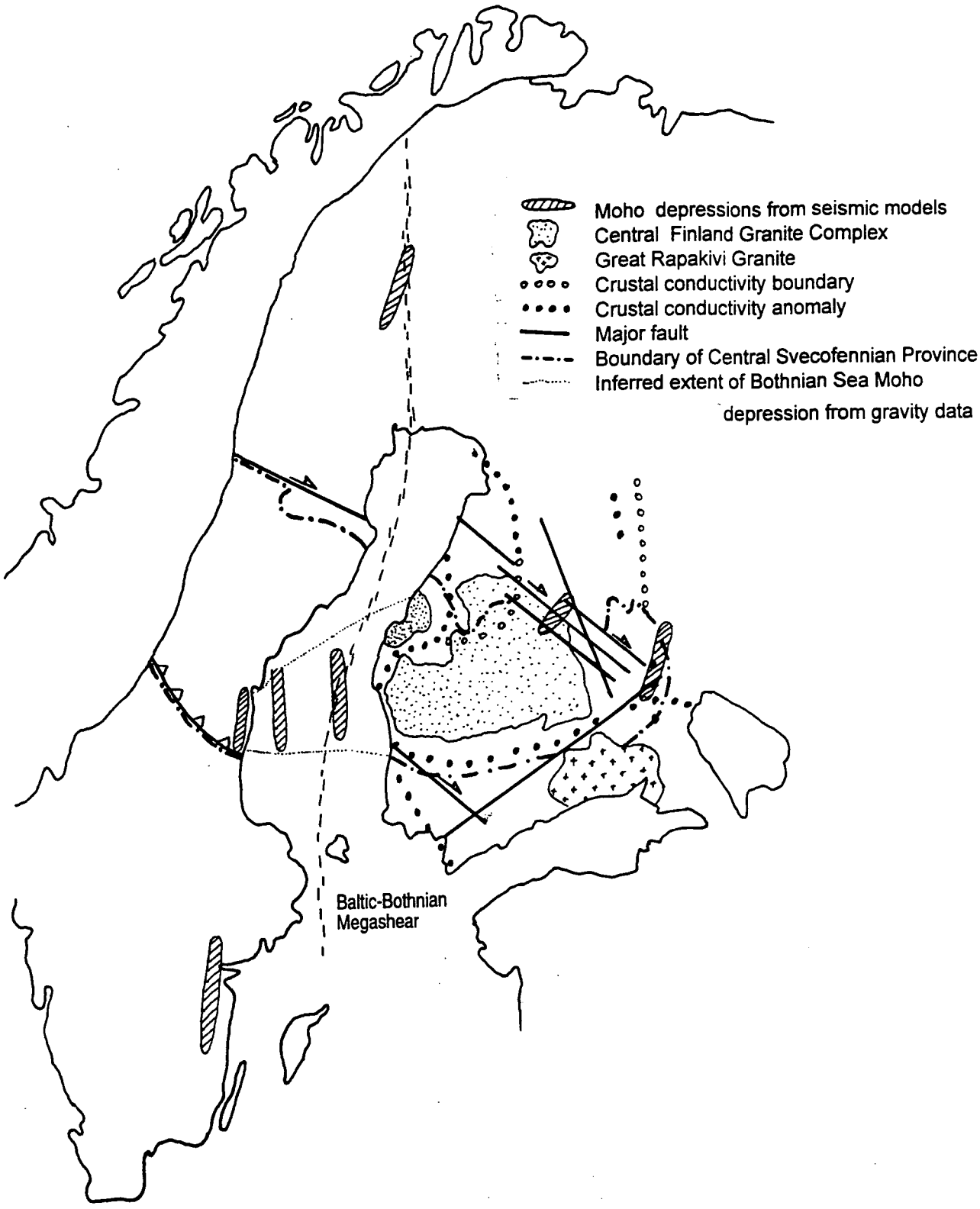


Fig. 8.18 Map showing synthesis of geological and geophysical information in the vicinity of the Bothnian Sea.

carbon-bearing sedimentary rocks of a closed oceanic basin (Korja and Koivukoski, 1994). The conductor has also been linked to the Storavan-Skelleftea anomaly in northern Sweden, marking the boundary between the Northern and Central Svecofennian terranes.

It is noticeable from the SVEKA magnetotelluric profile (Fig. 3.7a) that there are deep crustal conductive zones beneath both the Tampere Schist Belt anomaly and the Lagoda-Bothnian Bay shear zone. The Tampere schist belt anomaly extends beneath the Bothnian sea to the west and would appear to intersect BABEL Line 1 above the location of the Moho depression (Fig. 3.9). If the lower crustal conducting zone follows the same trend, the two features may be related.

Moho depressions have previously been associated with crustal scale fault zones (BALTIC, SVEKA profiles: Lagoda - Bothnian Bay shear zone; FENNOLORA shotpoint G: Baltic-Bothnian Megashear; FENNO-LORA shotpoint E: Ljusnan thrust zone). There are two possible faults coinciding with the Line 1 Moho depression: the southward continuation of the Baltic-Bothnian Megashear (Berthelsen and Marker, 1986) or the tectonic zone separating the Central and Southern Svecofennian Provinces in Sweden and Finland (Berthelsen and Marker, 1986, Strömberg, 1976). The latter has been used to explain the Moho depression modelled in the FENNOLORA profile between shotpoints D and E (Fig. 8.19). The main objection to the Baltic-Bothnian megashear theory is the lack of corroborating evidence from either Line 6 or Line 7, although Bruguier's model leaves room for speculation. The 3D interpretation of the Line 1 models (Chapter 7, Fig. 7.33) also suggests that the Moho dip is laterally limited to the west and does not join up with the feature beneath the

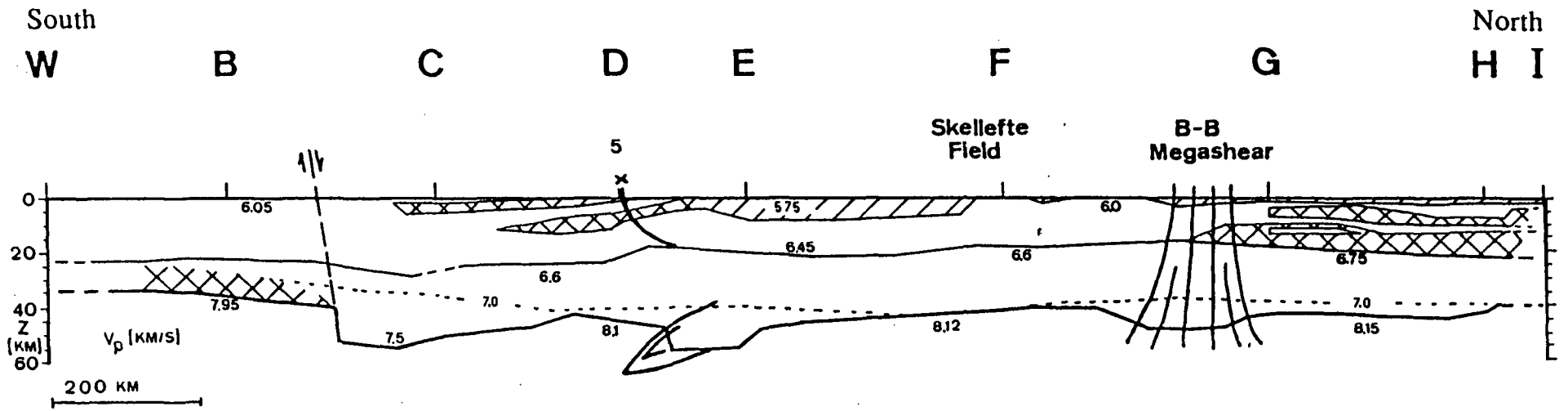


Fig. 8.19 Tectonic interpretation of FENNOLORA model (Guggisberg et al, 1991).

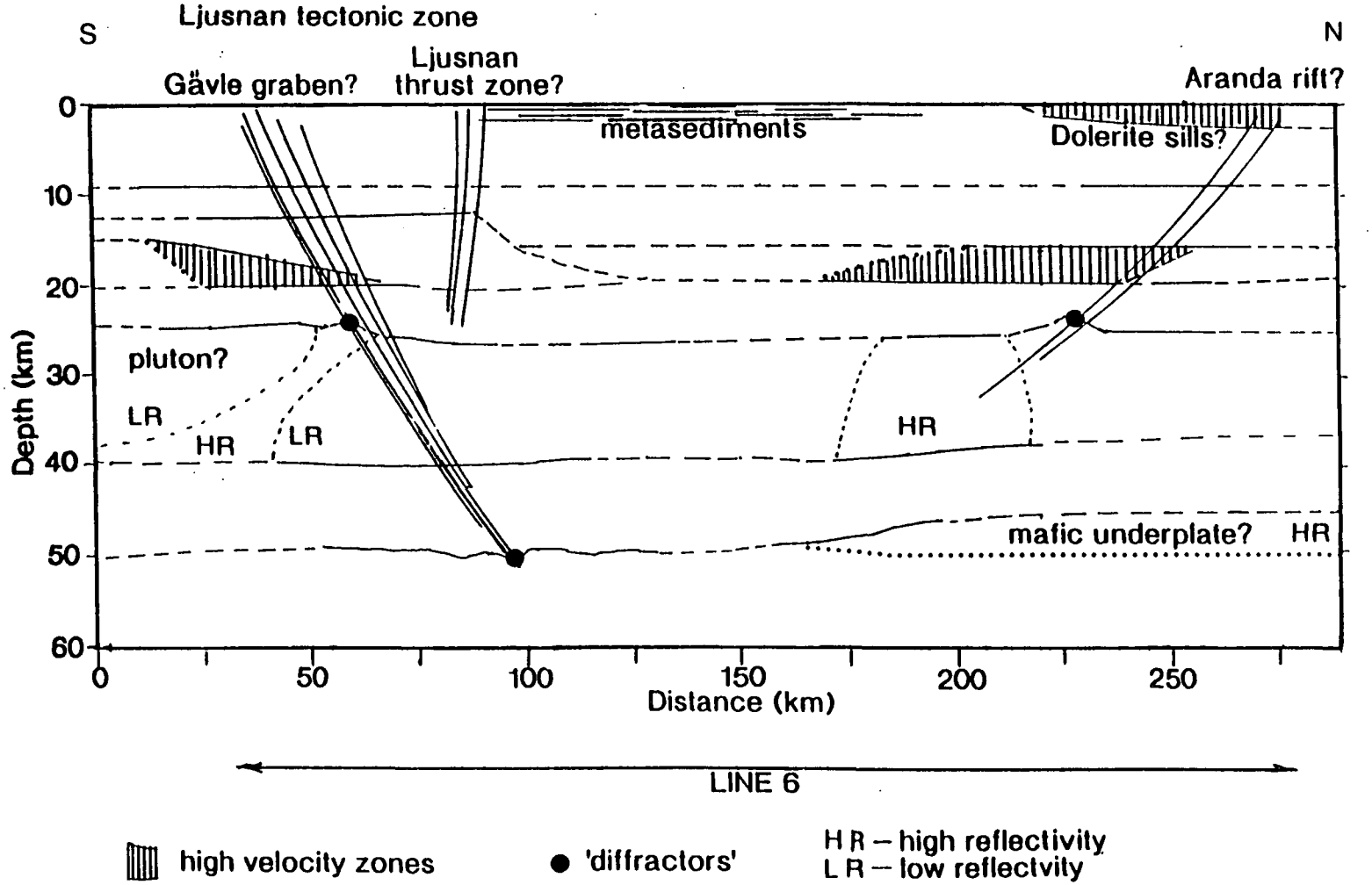


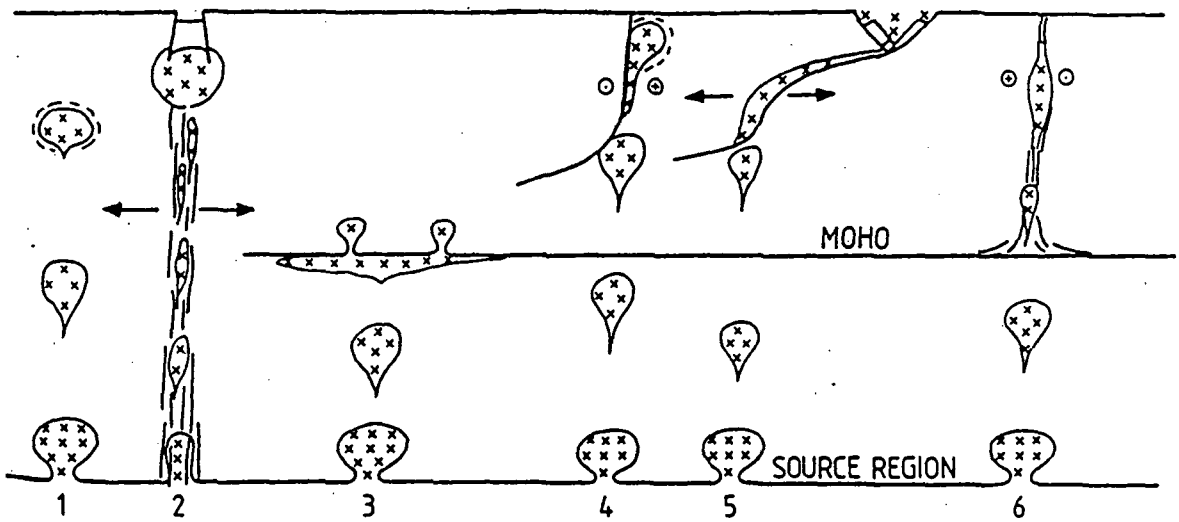
Fig. 8.20 Tectonic interpretation of BABEL Line 6 (Matthews, 1993).

FENNOLORA line, though this interpretation is only a first approximation which ignores the possibility of out of plane reflections.

The lack of a Moho depression in Matthews' Line 6 model affects the correlation of Line 1 with the FENNOLORA profile. However the crust in Line 6 is modelled as thinning to the North with a relatively low upper mantle velocity of 7.85 km s^{-1}. This is at odds with the other Baltic Shield models which have upper mantle velocities of at least 8.05 km s^{-1}. Matthews suggests that the highly reflective zone in the normal incidence data beneath the modelled Moho indicates a high velocity layer at the base of a thickened crust, not imaged in the wide angle data (Fig 8.20). This would enable the correlation of the Moho depressions beneath FENNOLORA and Line 1.

A further possibility is that the depression beneath Line 1 lies close to the intersection point of the Baltic - Bothnian Megashear and the southern boundary of the Central Svecofennian province.

The presence of a crustal scale fault may also explain the origin of the upper crustal high density body. Hutton (1988) and Hutton and Reavy (1992) have shown that large intrusive events may commonly be related to tectonic activity with a strike-slip component (transpressive or transtensional regimes). Hutton (1988) explains possible emplacement mechanisms for different tectonic regimes (Fig. 8.21), giving examples from around the globe. Hutton and Reavy (1992) give a more detailed model of emplacement along the Highland Boundary Fault during the Caledonian in Britain in which large volumes of magma with a high mantle component are generated at a Moho-detaching listric fault in a transpressional system (Fig 8.22). The crustal thickening associated with transpression is enhanced around the main and secondary faults giving



Six generalised modes of ascent and emplacement of granitoids; all begin with initial diapiric detachment and uprise of melts: (1) continued diapric uprise, in the absence of tectonics, leading to final arrest due to density equilibration followed by late ballooning; (2) uprise into major vertical tectonic extensional fault system, magmas rise to high levels, with uppermost crustal ponding and cauldron/caldera behaviour; (3) diapiric uprise arrested by viscosity/strength changes at Moho—this leads to lateral spreading with possible late spawning of crustal plutons; (4) diapiric rise into middle crust, intercepting an intra crustal strike slip fault zone leading to elongate plutons with late ballooning; (5) diapiric uprise intercepting intra crustal listric extensional fault/shear zone leading to listric granite sheets and possible generation of asymmetric cauldrons and calderas; (6) uprising melts intercept trans crustal vertical transcurrent fault/shear zone—jogs, pull apart and large tension gash features, etc., create space for magma ponding; note that in all these scenarios the "source region" is arbitrarily located in the lithospheric mantle.

Fig. 8.21 Six generalised modes of ascent and emplacement of granitoids, after Hutton, 1988

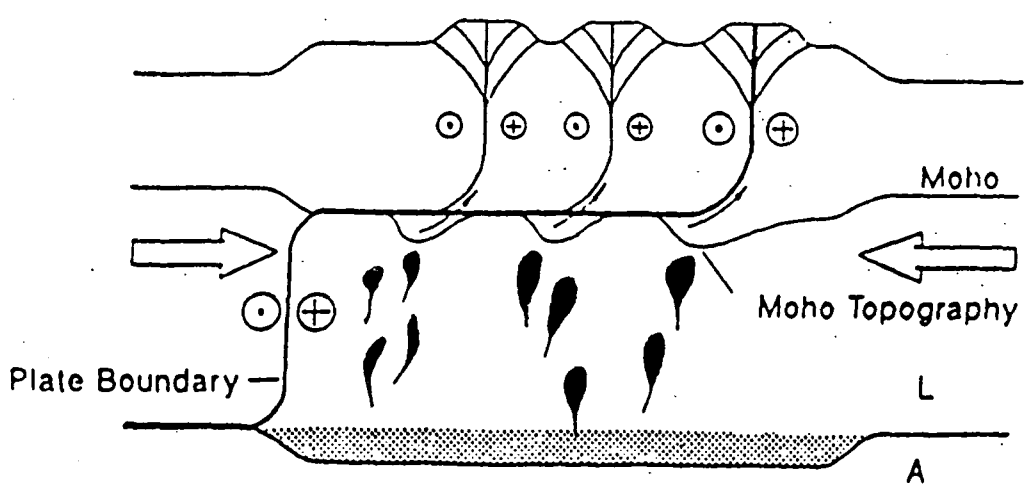


Fig. 8.22 A syntectonic igneous emplacement model for the Highland Boundary Faultsystem in the British Caledonides, after Hutton and Reavy, 1992

rise to Moho topography. The lithospheric mantle keel melts and rises to the Moho where continental crust is then melted, removing the Moho topography. Granitic magma, with a large mantle component, then rises up the shear zones which control ascent and emplacement. Although the Moho topography is removed in this model, the overall crustal thickening remains. The Kusznir-Matthews model, mentioned above, suggests a mechanism by which the topography may also be retained. The change in lower crustal properties across the depression (seismic velocity and density) may have an influence on its continued existence. Unfortunately, since there are no accessible igneous outcrops beneath the Bothnian Sea on which to look for strain indicators, this syntectonic emplacement model cannot be confirmed.

Fig. 8.23a shows a line drawing of the normal incidence data for Line 1 along with the reflectors from the synthetic section. The normal incidence data shows little structure which is of help in the tectonic interpretation, with patchy reflectivity and few reflectors greater than 10 km in length. However, changes in reflectivity between different parts of the profile suggest changes in structure. In Fig. 8.23b a tectonic model for Line 1 is suggested. The southern end of the profile is heavily intruded throughout the mid and lower crust, with both northward and southward dipping reflectors in the lower crust. The mid crust at the northern end is much less reflective but the normal incidence data shows evidence of steeply northward dipping reflectors which are interpreted as faults. The reflective lower crust at the northern end dips toward the south, ending approximately at the southern wall of the modelled Moho depression. The reflectivity at the base of the crust in the vicinity of the Moho depression decreases gradationally downwards. This system is interpreted, in a similar

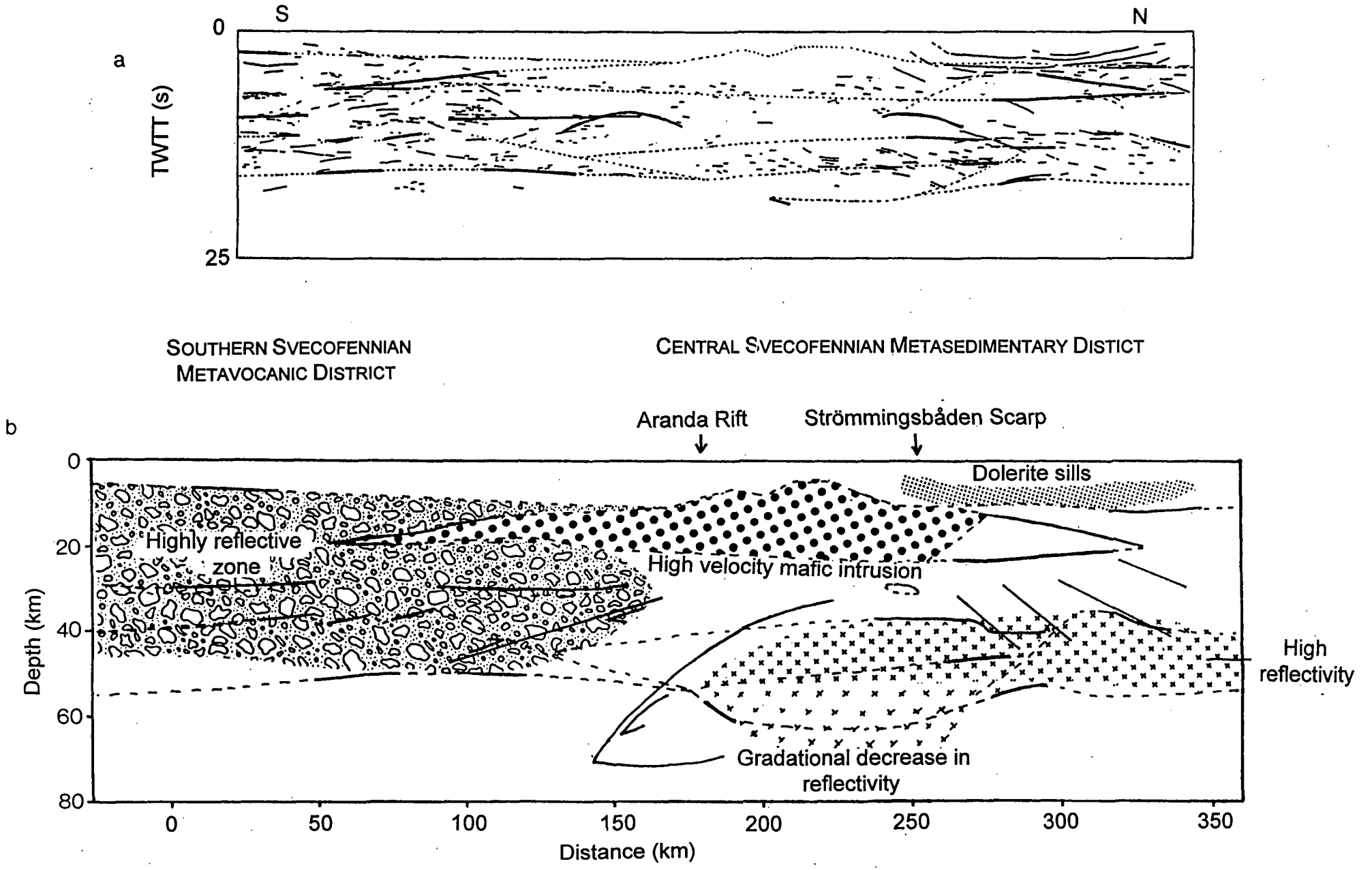


Fig. 8.23 a) Line drawing of Line 1 normal incidence data with normal incidence synthetic superimposed (bold reflections arise from parts of reflectors imaged at wide angle)

manner to the FENNOLORA line, as a crustal scale thrust fault, the southern section overriding the northern one. It is possible that the Baltic-Bothnian Megashear also cuts the section, the most probable location of the intersection being at 250 km along the line, near the Strömmingsbåden Scarp system. The only evidence for this is the disruption of reflectors in this region, the dipping reflectors which were interpreted above as faults, and the low velocity/density body in the upper crust.

8.7 Comparison with other regions

One of the initial aims of the BABEL project was to compare deep seismic information from a Precambrian shield region with that from Phanerozoic crust such as that around Britain and Western Europe. It should be noted before comparisons are made that the central part of Line 1 is probably atypical of Baltic Shield crust; the ends of the model are likely to be more representative.

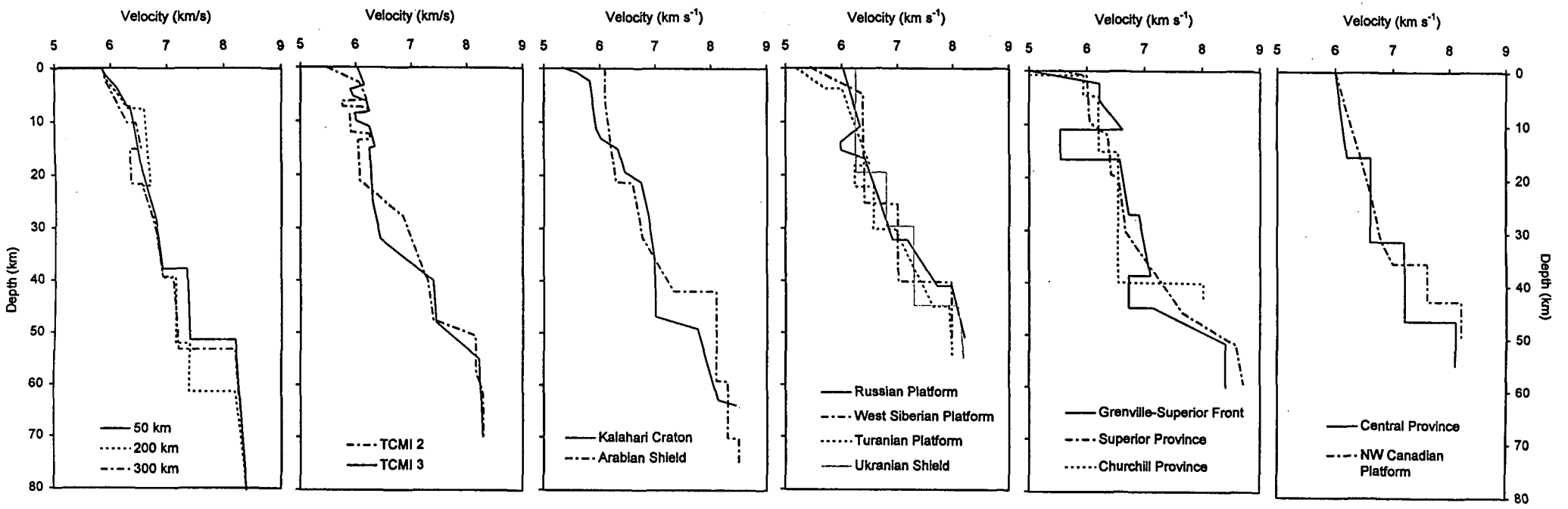
Phanerozoic continental crust has a mean thickness of around 28km and crustal velocity of 6.2 km s^{-1} (Meissner, 1986). Its bulk composition is that of andesite and it is overprinted by late extensional structures (Nelson, 1991). Normal incidence seismic data show a transparent upper crust overlying a reflective laminated lower crust. Various explanations exist for the cause of reflections from the lower crust such as mafic sills, trapped fluids, cumulate layering and ductile stretching fabric. Of these, the intrusion of mafic sills is currently believed to be the most plausible (Nelson, 1991, Mooney and Brocher, 1987). There is a sharp subhorizontal transition at the base of the lower crust to the unreflective upper mantle.

This boundary is widely regarded as coinciding with the wide angle reflection Moho.

Meissner (1986) calculates the mean thickness of Precambrian continental crust as 41 km with a crustal velocity of 6.6 km s^{-1}. The bulk composition is intermediate between andesite and basalt. Reflectivity patterns are complex and variable throughout the crust. The base of crustal reflectivity is sharp in some places, gradational in others and can change within the same region. Drummond and Collins (1986) and Durrheim and Mooney (1991) suggest that the increased crustal thickness may be due to basaltic underplating and that the preponderance of basaltic dike swarms contributes to the complex reflectivity patterns and lack of laminated reflectors. Durrheim and Mooney make an additional distinction between Archaean and Proterozoic crust, the latter being much thicker (average 45 km compared to 35 km for Archaean crust) and having a thicker high velocity layer at the base.

Pavlenkova (1979) and Nelson (1991) distinguish three general layers within Precambrian crust based on average velocities and velocity gradients. The middle layer has a lower gradient than the other two and is also more seismically transparent. Fig. 8.24 shows 1D velocity profiles from BABEL Line 1 compared with profiles from other Precambrian regions around the world.

The steep gradient and relatively high velocities in the uppermost 10 km of the BABEL Line 1 model bear most resemblance to the Russian and West Siberian Platforms and also the Grenville-Superior boundary in North America. High and low velocity layers are, with one exception, confined to the top 23 km. These are seen in the North Australian Craton, the Russian and Turanian Platforms and the Grenville-Superior Front.



TCMI 2/3, North Australian Craton	Finlayson, 1982
Kalahari Craton, SW Africa	Baier et al. 1983
Arabian Shield	Mooney et al. 1985
Russian Platform; W. Siberian Platform; Turanian Platform	Pavlenkova 1979
Ukranian Shield	Mooney et al. 1985
Grenville-Superior Front; Superior Province	Berry & Fuchs 1973
Churchill Province, N. America	Green et al. 1980
Central Province, N. America; NW Canadian Platform	Mooney & Braile 1989

Fig. 8.24 Velocity profiles from Precambrian crustal regions worldwide.

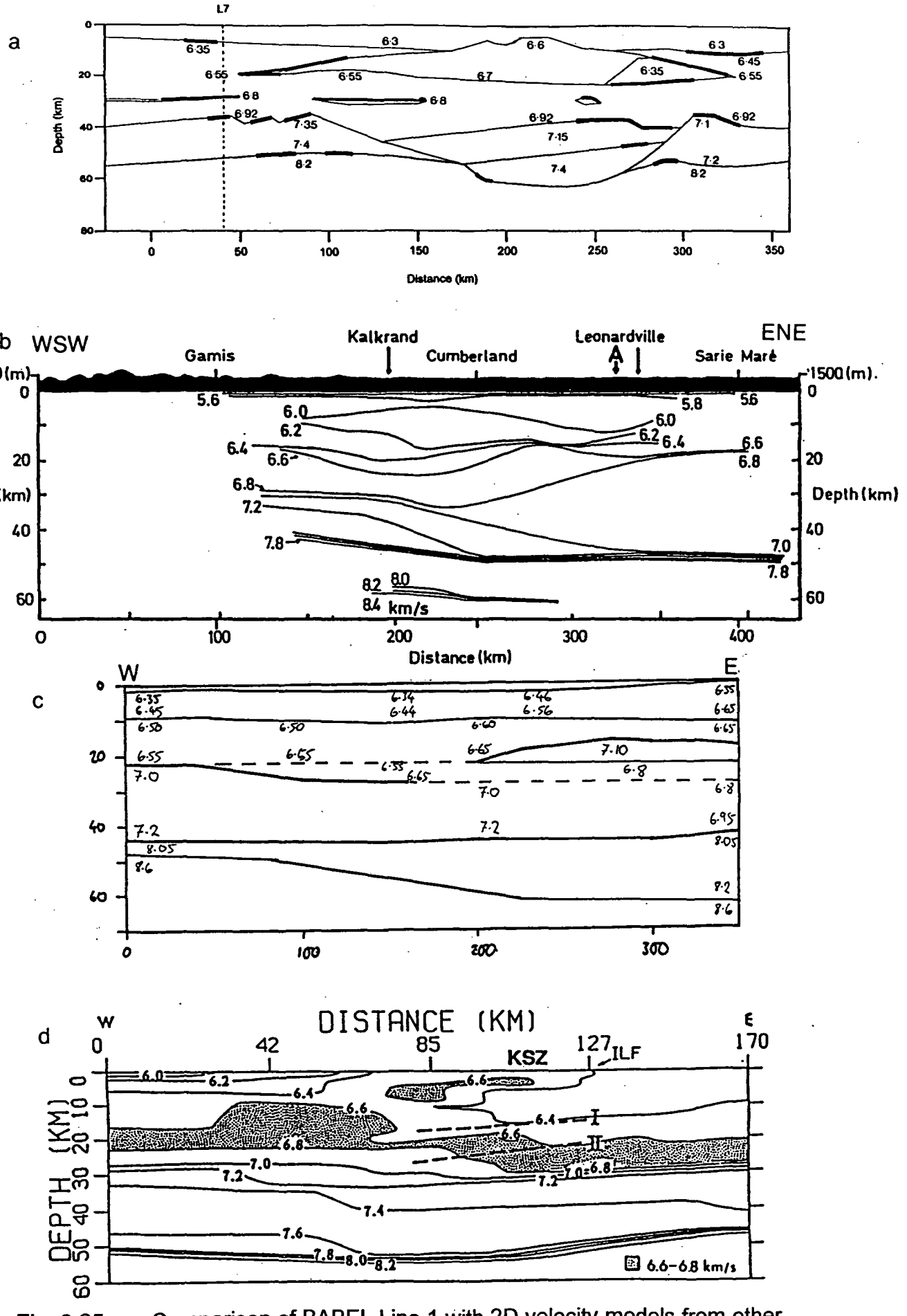


Fig. 8.25 Comparison of BABEL Line 1 with 2D velocity models from other Precambrian regions.

a) BABEL Line 1 Model A

b) Kalahari Craton (Baier et al. 1983)

c) SE Grenville Province (Hughes and Leutgert, 1992)

d) Kapuskasing Structural Zone, Superior Province (Clowes, 19

Between 18 and 25 km depth there is generally either a velocity step or a change in gradient, the velocity below exceeding 6.5 km s^{-1} in all cases except the North Australian Craton where a lower mid-crustal velocity is recorded.

Most of the profiles show a relatively featureless interval of low velocity gradient below this. The thickness of this zone varies from less than 10 km (N. Australian Craton profile TCMI 2, Turanian Platform) to more than 20 km (Kalahari Craton, Churchill Province of N. America). In BABEL Line 1 the zone is approximately 18 km thick.

A high velocity (V_p > 7 km s^{-1}) is present in all the profiles except for the Churchill Province and Kalahari Craton and even in the latter profile there is a transition layer to the Moho. Only at the Grenville-Superior front is a low velocity layer observed in the lower crust. The depth to the Moho is slightly greater for BABEL Line 1 than for most of the other profiles but shows a reasonable correlation with the North Australian Craton, Kalahari Craton and Superior Province.

In Fig 8.25, BABEL Line 1 Model A is compared with three other 2D models from Precambrian shield areas: the Kalahari Craton in southwest Africa (b), the southeastern Grenville Province (c) and the Kapuskasing Structural Zone in the Superior Province of North America (d). All three show strong lateral variations in velocity, including laterally limited high velocity layers in both North American profiles. The high velocity body at 20 km depth in the Grenville profile is interpreted as mafic cumulates, the remnant of magmatic intrusions. That in the Kapuskasing profile is thought to be due to a low angle thrust fault along which midcrustal rocks have been brought to the surface.

Moho topography is observed beneath the Kalahari Craton and Kapuskasing Structural Zone. In the first case this is associated with a lateral change in the crust while in the second it is associated with the Proterozoic age thrust fault mentioned above which extends down to the middle crust. This is further evidence that Moho topography can survive for long periods of geological time.

8.8 Conclusions

A large volume of high quality, high resolution seismic data was acquired in and around the Sea of Bothnia in the BABEL experiment of 1989 using marine airgun shots and land recording stations. The use of digital PDAS recording equipment proved partially successful, with the loss of a certain amount of data near the beginning, and thereafter requiring constant supervision. Once recorded, the data were much faster to process, by-passing the lengthy digitisation process necessary for the analogue Geostore data. The Geostores, on the other hand, allowed unsupervised recording of up to two days' worth of data in the field between tape changes. This enabled many stations to be operated by a small number of people.

Reversed 2D modelling of Line 1 was carried out for the incomplete in-line dataset, and unreversed modelling for two of the off-line datasets. These were combined to give an approximate three dimensional interpretation, gravity modelling providing an additional constraint to the in-line model. A number of interesting features were found, including a significant increase in crustal thickness extending for over 100 km in the central and northern part of the line. This is in agreement with Heikkinen

and Luosto's model derived from an independent dataset and can be tied in with the negative gravity anomaly at the northern end of the Bothnian Sea. A high velocity, high density upper crustal body interpreted as a dioritic pluton accounts for the small gravity high to the south of this.

Also notable are the non-smooth nature of the boundaries in the lower crust and the fragmented reflectors in the mid-crust. The mid-upper crust is found to be laterally inhomogeneous, with high and low velocity zones in certain areas. The major reflected phases are generally found to arise from relatively short boundary segments (<50 km) most of which can be tied in with short reflectors in the normal incidence data. This brings into doubt the existence of continuous boundaries over large areas, given the lack of evidence for them from the normal incidence data.

The models may provide evidence for the presence of a major tectonic shear zone (the Baltic-Bothnian Megashear), a crustal scale thrust fault, or both, in the central/northern part of the line. Although the evidence for either of these is not conclusive, there is clearly a crustal scale structure here.

The major drawbacks of the dataset, apart from the data loss at station 101P, were the lack of near surface information due to the large minimum offset between shot and receiver (>20 km), and the poor coverage towards the centre of the line due to its length. Both of these disadvantages could have been avoided by the deployment of one or more ocean bottom seismographs (OBS) along the line. This, however, would have necessitated the use of a second ship to deploy and recover the instruments, which funds did not allow. Nevertheless, it is a point to be borne in mind when planning future experiments.

8.9 Further work

There is a large quantity of data yet to be processed and studied from the off-line stations situated around the Swedish coast of the Sea of Bothnia. This could be used to find the limits and dimensions of the Moho depression more accurately. There is no obvious way in which to display and interpret this fan-spread data, although Heikkinen and Luosto (1992) have attempted to use normal moveout corrections to look at the Moho reflected arrival. The way forward must inevitably be integrated three dimensional modelling of all the relevant wide angle and normal incidence datasets. Perhaps a new way of defining the model should be used in which short length, isolated reflectors can be defined in a more or less smooth velocity gradient.

The data also contain much information from converted S-wave phases, mostly converted at the sea bed (Matthews, 1993). This information must also be included in the modelling process to provide valuable additional constraints.

References

Ahlberg, P., 1986. Den Svenska Kontinentalsockelns Beggrund. SGU Rpt. **47**.

Andersson, U. B., 1991. Granitoid episodes and mafic-felsic magma interaction in the Svecofennian of the Fennoscandian Shield, with main emphasis on the 1.8 Ga plutonics, Precamb. Res. **51**, 127-149.

Ansorge, J., Prodehl, C. and Bamford, D., 1982. Comparative interpretation of explosion seismic data, J. Geophysics, **51**, 69-84.

Axberg, P., 1981. Seismic stratigraphy and bedrock geology of the Bothnian Sea, N. Baltic. Act. Univ. Stockholm Contributions in Geology **36/3** p201.

BABEL Working Group 1990. Evidence for early Proterozoic plate tectonics from seismic reflection profiles in the Baltic Shield. Nature **138**, 34-38.

BABEL Working Group 1991. Reflectivity of a Proterozoic shield: examples from BABEL seismic profiles across Fennoscandia, Geodynamics series **22**, 77-86.

BABEL Working Group 1993. Integrated seismic studies of the Baltic shield using data in the Gulf of Bothnia region, Geophys. J. Int. **112**, 305-324.

Baier, B., Berckhemer, H., Gajewski, D., Green, R. W., Grimsel, Ch., Prodehl, C. And Vrees, R., 1983. Deep seismic sounding in the area of the Damara Orogen, Namibia, South West Africa, in Intracontinental Fold Belts, Springer, Berlin, pp 885-900, eds Martin, H. and Eder, F. W.

Barbey, P., Convert, J., Moreau, B., Capdevila, R. and Hameurt, J., 1984. Petrogenesis and evolution of an Early Proterozoic collisional orogen: the granulite belt of Lapland and the Belomorides (Fennoscandia), Bull. Geol. Soc. Finl. **56**, 161-188.

Berry, M. J. and Fuchs, K., 1973. Crustal structure of the Superior and Grenville provinces of the northeastern Canadian Shield, Bull Seismol. Soc. Am., **63**, 1393-1432.

Berthelsen, A. and Marker, M., 1986. 1.9-1.8 Ga old strike-slip megashears in the Baltic Shield, and their plate tectonic implications, Tectonophysics **128**, 163-181.

Braile, L. W., 1989 Crustal structure of the continental interior, in Geophysical framework of the continental United States, Geol. Soc. Am. Memoir **172** 285-315, eds Pakiser, L. C. And Mooney, W. D.

Bruguier, N., 1992. Crustal Structure of the southern Bothnian Sea from wide-angle reflection data: BABEL Line 7, Unpublished MSc. thesis, University of Durham.

Cassel, B. R. and Fuchs, K., 1979. Seismic investigations of the subcrustal lithosphere beneath Fennoscandia, J. Geophys. **46**, 369-384.

Cerveny, V., 1983. Synthetic body wave seismograms for laterally varying structures by the Gaussian beam method, Geophys. J. R. astr. Soc. **73**, 389-426.

Cerveny, V., 1985a. Ray synthetic seismograms for complex two-dimensional and three-dimensional structures, J. Geophys. **58**, 2-26.

Cerveny, V., 1985b. Gaussian beam synthetic seismograms, J. Geophys. **58**, 44-72.

Chapman, C. H., 1985. Ray theory and its extensions: WKBJ and Maslov seismograms, J. Geophys. **58**, 27-43.

Clowes, R.M., 1993. Variations in continental crustal structure in Canada from LITHOPROBE seismic reflection and other data, Tectonophysics **219**, 1-27.

Drummond, B. J. and Collins, C. D. N., 1986. Seismic evidence for underplating of the lower continental crust of Australia, Earth and Planetary Science Letters **79**, 361-372.

Durrheim, R. J. and Mooney, W. D., 1991. Archean and Proterozoic crustal evolution: Evidence from crustal seismology, Geology **19**, 606-609.

EUGENO-S Working Group, 1988. Crustal Structure and Tectonic evolution of the transition between the Baltic Shield and the North German Caledonides (the EUGENO-S project), Tectonophysics, **150**, 253-348.

Finlayson, D. M., 1982. Seismic crustal structure of the Proterozoic North Australian Craton between Tennant Creek and Mount Isa, J. Geophys. Res. **87**, 569-578.

Flodén, T. and Winterhalter, B., 1981. Pre-Quaternary geology of the Baltic Sea, in The Baltic Sea, pp 1-53, ed. Voipio, A., Elsevier.

Fuchs, K. and Müller, G., 1971. Computation of synthetic seismograms with the reflectivity method and comparison with observations, Geophys. J. R. astr. Soc. **23**, 417-433.

Gaál, G., 1986. 2200 million years of crustal evolution: the Baltic Shield, Bull. Geol. Soc. Finl. **58** (Part 1), 149-168.

Gaál, G., Mikkola, A. and Söderholm, B., 1978. Evolution of the Archaean crust in Finland, Precamb. Res. **6**, 199-215.

Gaál, G. and Gorbatschev, R., 1987. An outline of the Precambrian evolution of the Baltic shield, Precamb. Res. **35**, 15-52.

Gajewski, D., Psencik, I., 1986. Numerical modelling of seismic wave fields in 3-D laterally varying layered anisotropic structures - Program ANRAY86, Internal Report, Inst. of Earth and Planet. Phys., University of Alberta, Edmonton.

Gibson, B., S. and Levander, A., R., 1988. Lower crustal reflectivity patterns in wide-angle seismic recordings, Geophys. Res. Lett. **15(6)**, 612-620.

GLIMPCE seismic refraction working group, 1989. GLIMPCE seismic experiments: Long offset recording. EOS, **70**, 841-853.

Goleby, B. R., Kennet, B. L. N., Wright, C., Shaw, R. D. and Lambeck, K., 1990. Seismic reflection profiling in the Proterozoic Arunta Block, central Australia: processing for testing models of tectonic evolution, Tectonophysics **173**, 257-268.

Gorbunov, G. I., Zagorodny, V. G. and Robonen, E. I., 1985. Main features of the geological history of the Baltic Shield and the epochs of ore formation, Geol. Surv. Finl. Bull. **333**, 3-41.

Grad, M., and Luosto, U., 1987. Seismic models of the crust of the Baltic shield along the SVEKA profile in Finland, Annales Geophysicae **5B(6)**, 639-650.

Green, A. G., Stephenson, O. G., Mann, G. D., Kanasewich, E. R., Cumming, G. L., Hajnal, Z., Mair, J. A. And West, G. F., 1980.Cooperative surveys across the Superior-Churchill boundary zone in southern Canada, Can. J. Earth Sci. **17**, 617-632.

Guggisberg,B., Kaminski, W., and Prodehl, C., 1991. Crustal structure of the Fennoscandian Shield: A traveltime interpretation of the long-range FENNOLORA seismic refraction profile, Tectonophysics **195**, 105-137.

Heikkinen, P. and Luosto, U., 1992. Velocity structure and reflectivity of the Proterozoic crust in the Bothnian Sea, in The Babel Project : First Status Report, 1992, Commission of the European Communities, Directorate General, Science, Research and Development, Deep Reservoir Geology programme, pp 65-69, eds Meissner, R., Snyder, D., Balling, N. & Staroste, E.

Hjelt, S.-E., 1992. Geophysics in the BABEL/Bothnian Bay area: a preliminary compilation of non-seismic data, in The Babel Project : First Status Report, 1992, Commission of the European Communities, Directorate General, Science, Research and Development, Deep Reservoir Geology programme, pp 65-69, eds Meissner, R., Snyder, D., Balling, N. & Staroste, E.

Hjelt, S.-E., 1990. Electromagnetic studies on the EGT Northern Segment: summary of results, in Proceedings of the Sixth Workshop on the European Geotraverse (EGT) Project: Data Compilations and Synoptic Interpretation, 1989, Einsielden, pp. 53-63, eds Freeman, R & Mueller, St., European Science Foundation, Strasbourg.

Hughes, S. And Luetgert, J. H., 1992. Crustal Structure of the Southeastern Grenville Province, Northern New York State and Eastern Ontario, J. G. R. **97(B12)** 17455-17479.

Hutton, D. W. H., 1988. Granite emplacement mechanisms and tectonic controls: inferences from deformation studies, Trans. R. Soc. Edinburgh: Earth Sciences **79**, 245-255.

Hutton, D. W. H. and Reavy, R. J., 1992. Strike-slip tectonics and granite petrogenesis, Tectonics **11(5)**, 960-967.

Juhlin, C., 1990. Interpretation of the reflections in the Siljan Ring area based on results from the Gravberg-1 borehole, Tectonophysics **173**, 345-360.

Kennett, B. L. N. and Harding, A. J., 1985. Is ray theory adequate for reflection seismic modelling (A summary of reflection methods), First Break **3(1)**, 9-14.

Kontinen, A., 1987. An early Proterozoic ophialite - the Jormua mafic-ultramafic complex, northeastern Finland, Precamb. Res. **35**, 313-341.

Korja, T. and Koivukoski, K., 1994. Crustal conductors along the SVEKA profile in the Fennoscandian (Baltic) Shield, Finland, Geophys. J. Int., **116**, 173-197.

Kosminskaya, I. P., Belyaevsky, N. A. And Volvovsky, I. S., 1969. Explosion seismology in the USSR, in The Earth's Crust and Upper Mantle, Am. Geophys. Union, Geophys. Monogr. **13** 195-208, ed Hart, P.J.

Kröner, A., Puustinen, K. and Hickman, M., 1981. Geochronology and an Archean tonalitic gneiss dome in northern Finland and its relationship with an unusual overlying volcanic conglomerate and komatiitic greenstone, Contrib. Mineral. Petrol. **76**, 33-41.

Kusznir, N. J. and Matthews, D. H. , 1988. Deep seismic reflections and the deformational mechanics of the continental lithosphere, Journal of Petrology, Special lithosphere issue, 63-87.

Laajoki, K., 1986. The Precambrian supracrustal rocks of Finland and their tectono-exogenic evolution, Precamb. Res. **33**, 67-85.

Lewis, A. H. J., 1986. The deep seismic structure of Northern England and adjacent marine areas from the Caledonian Suture seismic project, Unpublished Ph.D. thesis, University of Durham.

Lund, C.-E., 1979. The fine structure of the lower lithosphere underneath the Blue Road profile in northern Scandinavia, Tectonophysics **56**, 111-122.

Lundberg, B., 1980. Aspects of the geology of the Skellefte field, northern Sweden. Geol. Fören. Stockholm Förh. **102**, 156-166.

Lundqvist, Th., 1987. Early Svecofennian stratigraphy of southern and central Norrland, Sweden, and the possible existence of an Archaean basement west of the Svecokarelides, Precamb. Res. **35**, 343-352.

Luosto, U., 1986. Reinterpretation of Sylen-Porvoo refraction data, Institute of Seismology, University of Helsinki, Report S-13, ed. Korhonen, H.

Luosto, U., 1990. Seismic data from the Northern segment of the EGT and from nearby profiles, in Proceedings of the Sixth Workshop on the European Geotraverse (EGT) Project: Data Compilations and Synoptic Interpretation, 1989, Einsielden, pp. 53-63, eds Freeman, R & Mueller, St., European Science Foundation, Strasbourg.

Luosto, U., Tiira, T., Korhonen, H., Azbel, I., Burmin, V., Buyanov, A., Kosminskaya, I., Ionkis, V. and Sharov, N., 1990. Crust and upper mantle structure along the DSS Baltic profile in S.E. Finland, Geophys. J. Int. **101**, 89-110.

McBride, J., H., Snyder, D., B. and Hobbs, R., W., 1992. Velocity analysis of BABEL marine reflection data, in The Babel Project : First Status Report, 1992, Commission of the European Communities, Directorate General, Science, Research and Development, Deep Reservoir Geology programme, pp 59-64, eds Meissner, R., Snyder, D., Balling, N. & Staroste, E.

Matthews, D. H., 1987. Can we see granites on seismic reflection profiles, Annales Geophysicae **5B** 353-356.

Matthews, P. A., 1993. Crustal Structure of the Baltic Shield beneath the Sea of Bothnia: BABEL Line 6, PhD Thesis, University of Durham

Matthews, P. A., Graham, D. P. and Long, R. E. , 1992. Crustal structure beneath the BABEL Line 6 from wide angle and normal incidence reflection data, in The Babel Project : First Status Report, 1992, Commission of the European Communities, Directorate General, Science, Research and Development, Deep Reservoir Geology programme, pp 101-104, eds Meissner, R., Snyder, D., Balling, N. & Staroste, E.

Meissner, R., 1986. The continental Crust, A Geophysical Approach, International Geophysics Series Volume 34, Academic Press, New York.

Mooney, W. D., Gettings, M. E., Blank, H. R. And Healy, J. H., 1985. Saudi Arabian seismic refraction profile: a traveltime interpretation of crustal and upper mantle structure, Tectonophysics **111**, 173-246.

Mooney, W. D. and Brocher, T. M., 1987. Coincident seismic reflection/refraction studies of the continental lithosphere: a global review, Geophys J. R. astr. Soc. **89**, 1-6.

Mooney, W. D. and Braile, L. W., 1989. The seismic structure of the continental crust and upper mantle of North America, in The Geology of North America - an Overview, Geol. Soc. Am., The Geology of North America **A**, 39, eds Bally, A. W. and Palmer, A. R.

Nelson, K. D., 1991. A unified view of craton evolution motivated by recent deep seismic reflection and refraction results, Geophys. J. Int. **105**, 25-35.

Pajunpää, K, 1987. Conductivity anomalies in the Baltic Shield in Finland, J. R. astr. Soc. **91**, 657-666.

Park, A. F., 1988. Nature of the early Proterozoic Outokumpu assemblage, eastern Finland, Precamb. Res. **38**, 131-146.

Pavlenkova, N. I., 1979. Generalized geophysical model and dynamic properties of the continental crust, Tectonophysics **59**, 381-390.

Pedersen, L. B., Tryggvason, A., Schmidt, J. and Gohl, K., 1992. Synthesis of geophysical data in the Bothnian sea, in The Babel Project : First Status Report, 1992, Commission of the European Communities, Directorate General, Science, Research and Development, Deep Reservoir Geology programme, pp 59-64, eds Meissner, R., Snyder, D., Balling, N. & Staroste, E.

Saverikko, M., 1983. The Kummitsoiva komatiite complex and its satellites in northern Finland, Bull. Geol. Soc. Finl. **55**, 111-139.

Sollogub, V. B., 1969. Seismic crustal studies in southeastern Europe, in The Earth's Crust and Upper Mantle, Am. Geophys. Union, Geophys. Monogr. **13**, 195-208, ed Hart, P.J.

Stromberg, A. G. B., 1976. A pattern of tectonic zones in the western part of the East European Platform, Geol. Fören. Stockholm Förh. **98**, 227-243.

Sundblad, K., 1991. Lead isotopic evidence for the origin of 1.8-1.4 Ga ores and granitoids in the southeastern part of the Fennoscandian Shield, Precamb. Res. **51**, 265-281.

Ward, P., 1987. Early Proterozoic deposition and deformation at the Karelian craton margin in southeastern Finland, Precamb. Res. **35**, 71-93.

Weber, M., 1988. Computation of body-wave seismograms in absorbing 2-D media using the Gaussian beam method: comparison with exact methods, Geophysical J. **92**, 9-24.

Welin, E., 1987. The depositional evolution of the Svecofennian supracrustal sequence in Finland and Sweden, Precamb. Res. **35**, 95-113.

Welin, E. and Stålhös, G., 1986. Maximum age of the synmetamorphic Svecokarelian fold phases in south central Sweden, Geol. Fören. Stockholm Förh. **108**, 31-34.

Wellman, P., 1988. Development of the Australian Proterozoic crust as inferred from gravity and magnetic anomalies, Precamb. Res. **40**, 89-100.

West, T. E., 1990. A high resolution wide-angle seismic study of the crust beneath the Northumberland trough, Unpublished Ph.D. thesis, University of Durham.

Winterhalter, B., 1972. On the geology of the Bothnian Sea, an epeiric sea that has undergone Pleistocene Glaciation, Geol. Surv. Fin. Bull. **258**.

Witschard, F., 1984. The geological and tectonic evolution of the Precambrian of northern Sweden - a case for basement reactivation?, Precamb. Res. **23**, 273-315.

Appendix A: Computer programs

Contents:

1. BPFILT

```
      program bpfilt
c
c accepts trace file sequence and band pass filters
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      integer*4 ierr, ihand, ohand, recsz
      integer*4 param
      integer*4 itrace, ifirst, ilast, nrec
      integer*4 iwopt,nsampl
      real*4    work(16384),srate,f1,f2,f3,f4
      character iname*20, oname*24, ipath*25, opath*25
c
      call dlsini()
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c defaults
c
      ipath = './                '
      opath = './                '
      iname = '1_160bs5          '
      oname = '                  '
c
c     window is hanning
      iwopt = 2
c
      f1 = 3.0
      f2 = 4.0
      f3 = 30.0
      f4 = 40.0
c
c ifirst is trace no. at start of loop, ilast is
c trace no. at end, and nrec is step size
      ifirst = 1
      ilast  = 9999
      nrec   = 1
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c get parameter values from user via menu
c
  1   call menu(ipath,opath,iname,iwopt,f1,f2,f3,f4,ifirst,ilast)
      print*
      print 4
      read*,param
      print*
      if (param.EQ.99) goto 9
      if (param.EQ.1) then
            print*,'enter new input pathname :'
            read 5,ipath
      elseif (param.EQ.2) then
            print*,'enter new output pathname :'
            read 5,opath
      elseif (param.EQ.3) then
            print*,'enter new filename :'
            read 5,iname
      elseif (param.EQ.4) then
            print*,'enter window type :'
            read 6,iwopt
      elseif (param.EQ.5) then
            print*,'enter low-cut frequency (bottom) :'
            read 7,f1
      elseif (param.EQ.6) then
            print*,'enter low-cut frequency (top) :'
```

```
            read 7,f2
         elseif (param.EQ.7) then
            print*,'enter high-cut frequency (top) :'
            read 7,f3
         elseif (param.EQ.8) then
            print*,'enter high-cut frequency (bottom) :'
            read 7,f4
         elseif (param.EQ.9) then
            print*,'enter first trace number :'
            read 8,ifirst
         elseif (param.EQ.10) then
            print*,'enter last trace number :'
            read 8,ilast
         endif
         go to 1
c
   4     format (2X,'Enter number of parameter to alter (99 to continue) : ',$)
   5     format (A)
   6     format (I)
   7     format (F5.0)
   8     format (I4)
c
   9     continue
c
         oname = iname(1:lnb(iname))//'_bpf'
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c open input and output files
c
         call optdat (ipath,iname,0,recsz,ihand,ierr)
c
         if(ierr.LT.0) then
            call report('segy','dataset','F',
     &  ' Cannot find all dataset files')
            call exit(1)
         endif
c
      call optdat (opath,oname,1,recsz,ohand,ierr)
c
         if(ierr.LT.0) then
            call report('segy','dataset','F',
     &   ' Unable to create dataset - space? or exists?')
            call exit(1)
         endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c read segy header
c
         call rhead(ihand,0,ierr)
c
         if (ierr.LT.0) then
            call report('segy','dataset','F',
     &   ' Unable to read reel header file')
            call exit(1)
         endif
c
         nsampl = rhinsm
         if (rhisr.NE.0) srate = 1.0E6/rhisr
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      do 10 itrace=ifirst,ilast,nrec
            print 1000, itrace
 1000  format(1X,'Filtering trace : ',I5)
c
c
```

```
c read in trace header to common block
c
          call rtdh(ihand,itrace,ierr)
          if (ierr.LT.0) then
           call report('segy','dataset','F',
   &   ' Unable to read trace header file')
           call exit(1)
          endif
c
c check for end of file
c
          if (ierr.GT.0) then
           call report('segy','dataset','I',
   &   ' End of file reached')
           goto 11
          endif
c
          if (unsamp.NE.0) nsampl = unsamp
          if (wsr.NE.0) srate = 1.0E6/wsr
c
c read in data, apply filter + write to new file
c
          call rtrd (ihand,itrace,work,nsampl,ierr)
c
          if (ierr.LT.0) then
           call report('segy','dataset','F',
   &   ' Unable to read trace data file ')
           call exit(1)
          endif
c
      call bpass(nsampl,work,f1,f2,f3,f4,srate,iwopt)
c
          call wtrd (ohand,itrace,work,nsampl,ierr)
c
          if (ierr.LT.0) then
           call report('segy','trace data','F',
   &   ' Unable to write to file')
           call cltdat(ohand)
           call exit(1)
          endif
c
c put filter values to common block variables
c and write to new header file
c
          ulofrq = f1
          uhifrq = f4
c
c fill in some other variables required by plot programe
          if (ufldfn.EQ.0) ufldfn = ufldtn
          if (utrace.EQ.0) utrace = 1
c
          ushot =0
c
          call wtdh(ohand,itrace,ierr)
          if (ierr.LT.0) then
           call report('segy','trace data','F',
   &   ' Unable to write to file')
           call cltdat(ohand)
           call exit(1)
          endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
 10   continue
 11   continue
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
c print last trace header common block values
c
     print 100,ushot ,utrc  ,ufldfn,ufldtn,usrcpn,ucdp,
    &       utrace,utrtyp,unsum ,unstk ,uduse ,
    &       wrange,wrcvge,wsrcse,wsrcdp,wrcvde,wsrcde,
    &                   wsrcwd,wrcvwd,
    &                   wsrcx ,wsrcy ,wrcvx ,wrcvy ,ucunit,
    &                   wvelw ,wvelsw,usrcut,urcvut,usrcst,urcvst,
    &                   utotst,ulaga ,ulagb ,udelt ,umutst,umutfn,
    &                   unsamp,wsr   ,ugntyp,wgain ,wgnst ,
    &                   ucorr ,ufrqst,ufrqfn,uslen ,ustyp ,ustlen,
    &                   uetlen,utptyp,ualfrq,ualslp,unofrq,unoslp,
    &                   ulofrq,uhifrq,uloslp,uhislp,uyear ,uday  ,
    &                   uhour ,umin  ,usec  ,utctyp,xmsec ,
    &                   xeyear ,xeday  ,xehour ,xemin  ,xesec  ,xemsec ,
    &       xdelt ,xrvel,xrtoff ,xstncd,xevecd
c
 100  format(6(I10),/,5(I10),/,6(F10.2),/,2(F10.2),/,4(F10.2),I10,/,
    &       2(F10.2),4(I10),/,6(I10),/,I10,F10.2,I10,2(F10.2),/,
    &       3(6(I10),/),5(I10),/,6(I10),/,3(I10),2(6X,A4),/)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c write out file header with extended name
c
      call whead(ohand,0,ierr)
      if (ierr.LT.0) then
              call report('segy','dataset','F',
    &    ' Unable to write reel header file')
              call exit(1)
      endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c dump common block rhicom + extras
c
          print*,'\n Rhicom common block'
          print 101,rhintr,rhinax,rhisr ,rhisrf,
    &           rhinsm,rhinsf,ifmt  ,rhirli,rhilni, rhimes
  101  format(1X,3(I8),/,1X,6(I8))
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      call cltdat(ihand)
      call cltdat(ohand)
c
      stop
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
          subroutine menu(pathi,patho,name,iwopt,f1,f2,f3,f4,ifirst,ilast)
c
          character*(*) pathi, patho, name
          integer iwopt, ifirst, ilast
          real f1,f2,f3,f4
c
          print 200
          print 201,pathi
          print 202,patho
          print 203,name
          print 204,iwopt
          print 205,f1
          print 206,f2
          print 207,f3
          print 208,f4
          print 209,ifirst
          print 210,ilast
```

```
      print 299
c
 200  format(10X,'********************',/,
     &   10X,'* BAND PASS FILTER *',/,
     &   10X,'********************')
 201  format(2X,'1. Input pathname  :',T55,A25)
 202  format(2X,'2. Output pathname :',T55,A25)
 203  format(2X,'3. Filename        :',T55,A20)
 204  format(2X,'4. Window type (0=box,1=trapezoid,2=hanning)
:',T58,I1)
 205  format(2X,'5. Low-cut frequency  - bottom             :',T55,F5.1)
 206  format(2X,'6. Low-cut frequency  - top {not used for}  :',T55,F5.1)
 207  format(2X,'7. High-cut frequency - top { box window }  :',T55,F5.1)
 208  format(2X,'8. High-cut frequency - bottom             :',T55,F5.1)
 209  format(2X,'9. First trace number     :', T55,I4)
 210  format(1X,'10. Last trace number      :',T55,I4)
 299  format(1X,'99.            Run ')
c
      return
      end
```

2. COMBINE9

```
      program combine9
c
c combines up to 9 data and header files
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      integer*4 ierr, ihand(9), ohand, recsz
      integer*4 param, nsampl, filenum, maxnum
      integer*4 itrace, ifirst, ilast, nrec, otrace
      character*24 iname(9), oname, ipath
      real*4    work(16384)
c
      call dlsini()
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c defaults
c
      ipath  = './                '
      iname(1) = '1_l61bs2            '
      iname(2) = '2_l61bs2            '
      iname(3) = '3_l61bs2            '
      iname(4) = '4_l61bs2            '
      iname(5) = '5_l61bs2            '
      iname(6) = '6_l61bs2            '
      iname(7) = '7_l61bs2            '
      iname(8) = '8_l61bs2            '
      iname(9) = '9_l61bs2            '
      oname  = 'l61bs2              '
c
c ifirst is trace no. at start of loop, ilast is
c trace no. at end, and nrec is step size
      otrace = 1
      ifirst = 1
      ilast  = 9999
      nrec   = 1
      filenum= 1
      maxnum = 2
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c get parameter values from user via menu
c
  1   call menu(ipath,iname,maxnum,oname)
      print*
      print 4
      read*,param
      print*
      if (param.EQ.99) goto 9
      if (param.EQ.1) then
            print*,'enter new pathname :'
            read 5,ipath
      elseif (param.EQ.2) then
            print*,'enter number of files to combine :'
            read 6,maxnum
      elseif (param.EQ.3) then
            print*,'enter 1st filename :'
            read 5,iname(1)
      elseif (param.EQ.4) then
            print*,'enter 2nd filename :'
            read 5,iname(2)
      elseif (param.EQ.5) then
            print*,'enter 3rd filename :'
            read 5,iname(3)
```

```
      elseif (param.EQ.6) then
            print*,'enter 4th filename :'
            read 5,iname(4)
      elseif (param.EQ.7) then
            print*,'enter 5th filename :'
            read 5,iname(5)
      elseif (param.EQ.8) then
            print*,'enter 6th filename :'
            read 5,iname(6)
      elseif (param.EQ.9) then
            print*,'enter 7th filename :'
            read 5,iname(7)
      elseif (param.EQ.10) then
            print*,'enter 8th filename :'
            read 5,iname(8)
      elseif (param.EQ.11) then
            print*,'enter 9th filename :'
            read 5,iname(9)
      elseif (param.EQ.12) then
            print*,'enter output filename :'
            read 5,oname
      endif
      go to 1
c
   4  format (2X,'Enter number of parameter to alter (99 to continue) : ',$)
   5  format (A)
   6  format (I1)
c
   9  continue
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
  10  continue
c
```

```
c open input file, read segy header
      call optdat (ipath,iname(filenum),0,recsz,ihand(filenum),ierr)
c
      if(ierr.LT.0) then
        call report('segy','dataset','F',
     &  ' Cannot find all dataset files')
        call exit(1)
      endif
c
      call rhead(ihand(filenum),0,ierr)
c
      if (ierr.LT.0) then
        call report('segy','dataset','F',
     &  ' Unable to read reel header file')
        call exit(1)
      endif
c
      nsampl = rhinsm
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c open output files 1st time round
c
    if (filenum.eq.1) then
      call optdat (ipath,oname,1,recsz,ohand,ierr)
        if(ierr.LT.0) then
          call report('segy','dataset','F',
     &    ' Unable to create dataset - space? or exists?')
          call exit(1)
        endif
      endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
```

```
      do 100 itrace=ifirst,ilast,nrec
            print 1000,itrace,otrace
c
c read in trace header to common block
            call rtdh(ihand(filenum),itrace,ierr)
            if (ierr.LT.0) then
              call report('segy','dataset','F',
     &    ' Unable to read trace header file')
              call exit(1)
            endif
c
c check for end of file
            if (ierr.GT.0) then
              call report('segy','dataset','I',
     &     ' End of file reached')
           call cltdat(ihand(filenum))
              filenum=filenum+1
           if (filenum.LE.maxnum) then
                    goto 10
              else
                    goto 101
              endif
            endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c read in data and write to new file
c
           call rtrd(ihand(filenum),itrace,work,nsampl,ierr)
           call wtrd(ohand,otrace,work,nsampl,ierr)
c
c edit header
           ufldfn=otrace
           ufldtn=otrace
           if (xrvel.eq.0) xrvel=10000
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
            call wtdh(ohand,otrace,ierr)
            if (ierr.LT.0) then
             call report('segy','trace data','F',
     &    ' Unable to write to file')
              call cltdat(ohand)
              call exit(1)
            endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
           otrace=otrace+1
 100   continue
 101   continue
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c write out file header with extended name
c
      call whead(ohand,0,ierr)
      if (ierr.LT.0) then
            call report('segy','dataset','F',
     &    ' Unable to write reel header file')
            call exit(1)
      endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      call cltdat(ohand)
c
 1000  format(1X,' trace : ',I5,' ->',I5)
```

```
c
      stop
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
            subroutine menu(path,iname,maxnum,oname)
c
            integer maxnum
            character*(*) path, iname(maxnum), oname
c
            print 200
            print 201,path
            print 202,maxnum
            do i=1,maxnum
             print 203,i+2,i,iname(i)
            enddo
            print 204,oname
            print 299
c
 200        format(10X,'*****************',/,
     &        10X,'* COMBINE FILES *',/,
     &        10X,'*****************')
 201        format(2X,'1. Input pathname  :',T45,A25)
 202        format(2X,'2. Number of files :',T45,I1)
 203        format(2X,I1,'. Filename ',I1,'  :',T45,A20)
 204        format(2X,'12. Output filename :',T45,A20)
 299        format(1X,'99.            Run ')
c
            return
            end
```

3. EXTR

```
      program extr
c
c  accepts trace file sequence and extracts portion with optional resampling
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      integer*4 ierr, ihand, recsz, nsampl, itrace
      integer*4 ohand, new_recsz, new_nsampl, otrace
      integer*4 ifirst, ilast, nrec, tfirst,tlast
      integer*4 param
      character iname*32, oname*32, ipath*32
           real*4   intrace(16384), work(16384), outtrace(16384)
           real*4     srate,new_srate
           logical    resmp
c
      call dlsini()
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c defaults
           ipath = './'
           iname = 'l7bs4'
           oname = 'extract'
c
c ifirst is trace no. at start of loop, ilast is
c trace no. at end
       ifirst = 1
       ilast  = 9999
           nrec   = 1
           tfirst = 1
           tlast  = 4000
       itrace = 1
           otrace = 0
           resmp  = .false.
           new_srate=100.0
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c get parameter values from user via menu
c
      do while (param.ne.99)
           call menu(ipath,iname,oname,ifirst,ilast,nrec,tfirst,tlast,
     &                          resmp,new_srate)
           print*
           print 4
           read*,param
           print*
           if (param.EQ.1) then
                    print*,'enter new  pathname :'
                    read 5,ipath
           elseif (param.EQ.2) then
                    print*,'enter new input filename :'
                    read 5,iname
           elseif (param.EQ.3) then
                    print*,'enter new output filename :'
                    read 5,oname
           elseif (param.EQ.4) then
                    print*,'enter first trace number :'
                    read 8,ifirst
           elseif (param.EQ.5) then
                    print*,'enter last trace number :'
                    read 8,ilast
           elseif (param.EQ.6) then
                    print*,'enter trace step :'
                    read 8,nrec
```

```
      elseif (param.EQ.7) then
              print*,'enter first sample number :'
              read 8,tfirst
      elseif (param.EQ.8) then
              print*,'enter last sample number :'
              read 8,tlast
      elseif (param.EQ.9) then
              if (resmp) then
                resmp=.false.
              else
                resmp=.true.
              endif
      elseif ((param.EQ.10).and.(resmp)) then
              print*,'enter new sample rate  :'
              read 6,new_srate
      elseif (param.EQ.90) then
              call report('extract','menu','I',
     &        'user requested abort')
              call exit(1)
      endif
    enddo
c
      new_nsampl=tlast-tfirst+1
      new_recsz =new_nsampl*4
c
   4  format (2X,'Enter number of parameter to alter (99 to continue) : ',$)
   5  format (A)
   6  format (F5.0)
   8  format (I4)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c open  files and read segy header
c input
      call optdat (ipath,iname,0,recsz,ihand,ierr)
      if(ierr.LT.0) then
        call report('extract','dataset','F',
     &  ' Cannot find all dataset files')
        call exit(1)
      endif
c
      call rhead(ihand,0,ierr)
      if (ierr.LT.0) then
        call report('extract','dataset','F',
     &  ' Unable to read reel header file')
        call exit(1)
      endif
c
      nsampl = rhinsm
      if (rhisr.NE.0) srate  = 1.0E6/rhisr
c output
      call optdat (ipath,oname,1,new_recsz,ohand,ierr)
      if(ierr.LT.0) then
        call report('extract','dataset','F',
     &  ' Cannot find all dataset files')
        call exit(1)
      endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c MAIN LOOP
c
    do 10 itrace=ifirst,ilast,nrec
      otrace=otrace+1
c
      call razout('Trace : ',itrace)
      call flush(6)
c
```

```
c read in trace header to common block
          call rtdh(ihand,itrace,ierr)
          if (ierr.LT.0) then
           call report('extract','dataset','F',
   &    ' Unable to read trace header file')
           call exit(1)
          endif
c
c check for end of file
          if (ierr.GT.0) then
           call report('extract','dataset','I',
   &    ' End of file reached')
           goto 11
          endif
c
c check for correct sample no. and rate
          if (unsamp.NE.0) nsampl = unsamp
          if (wsr.NE.0) srate = 1.0E6/wsr
c
c read in data
          call rtrd (ihand,itrace,intrace,nsampl,ierr)
c
          if (ierr.LT.0) then
           call report('extract','dataset','F',
   &    ' Unable to read trace data file ')
           call exit(1)
          endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
          if (resmp) then
           call taper(nsampl,intrace,int(0.25*srate))
           call resamp(16384,intrace,16384,work,srate,new_srate,6)
           wsr=1.0E6/new_srate
           call aacopy(work,tfirst,tlast,outtrace,1,new_nsampl)
          else
           call aacopy(intrace,tfirst,tlast,outtrace,1,new_nsampl)
          endif
c
          unsamp=new_nsampl
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c write to file
          ufldfn=otrace
          ufldtn=otrace
          xrtoff=xrtoff+real(tfirst-1)*wsr/1.0E3
c
          call wtdh(ohand,otrace,ierr)
          if (ierr.LT.0) then
           call report('extract','dataset','F',
   &    ' Unable to write to trace header file')
           call exit(1)
          endif
c
          call wtrd (ohand,otrace,outtrace,unsamp,ierr)
          if (ierr.LT.0) then
           call report('extract','trace data','F',
   &    ' Unable to write to file')
           call cltdat(ihand)
           call exit(1)
          endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
  10  continue
c END OF MAIN LOOP
c
```

```
 11  continue
c
c write new file header, updating appropriate variables
          rhinsf=rhinsm
          rhinsm=unsamp
          rhisrf=rhisr
          rhisr=int(wsr)

          call whead(ohand,0,ierr)
          if (ierr.LT.0) then
            call report('extract','dataset','F',
     &    ' Unable to write to trace header file')
            call exit(1)
          endif
c
      call cltdat(ihand)
      call cltdat(ohand)
c
      stop
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
          subroutine menu(pathi,iname,oname,ifirst,ilast,nrec,tfirst,tlast,
     &                          resmp,new_srate)
c
          character*(*) pathi, iname, oname
          integer  ifirst, ilast, nrec, tfirst,tlast
          real*4     new_srate
          logical    resmp
c
          print*,'\f'
          print 200
          print 201,pathi
          print 202,iname
          print 203,oname
          print 204,ifirst
          print 205,ilast
          print 206,nrec
          print 207,tfirst
          print 208,tlast
          print 209,resmp
          if(resmp) then
           print 210,new_srate
          endif
          print 290
          print 299
c
  200     format(9X,'****************',/,
     &      9X,'* EXTRACT DATA *',/,
     &      9X,'****************')
  201     format(2X,'1. Input pathname       :',T45,A25)
  202     format(2X,'2. Input filename       :',T45,A20)
  203     format(2X,'3. Output filename      :',T45,A20)
  204     format(2X,'4. First trace number          :',T45,I4)
  205     format(2X,'5. Last trace number           :',T45,I4)
  206     format(2X,'6. Trace step                  :',T45,I4)
  207     format(2X,'7. First sample number         :',T45,I4)
  208     format(2X,'8. Last sample number          :',T45,I4)
  209     format(2X,'9. Resample ?                  :',T47,L1)
  210     format(1X,'10. New sample rate (Hz)       :',T43,F6.1)
  290     format(1X,'\n',1X,'90            ABORT')
  299     format(1X,'\n',1X,'99.            RUN ')
c
          return
          end
```

4. F-K

F-K.H

```
c
c       common block for variables in f-k program
c
        integer*4 param
        integer*4 ifirst, toff, pltype, nclass
c
        real*4    trsep, class(20), f_filt(4), v_filt(4), redvel
c
        character iname*32, oname*32, ipath*48
        character driver*16, title*24, vflag*4, fflag*4
c
        logical reversed, war
c
        common /fkcom/
     &          param, ifirst, toff, pltype, nclass, class, f_filt,
     &          v_filt, redvel, trsep, iname, oname, ipath,
     &          driver, title, vflag, fflag, reversed, war
        save /fkcom/
c
```

```
      program f_k
c
c f-k spectrum and velocity filtering
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      include "/home/kanga/dgl3dpg/fortran/include/f-k.h"
c
            integer*4 ihand, ohand, ierr, recsz, length
            integer*4 itrace, otrace, nsampl, size
c size  = data array dimensions - must be a power of 2
            parameter(size=512)
            real*4    buffer(16384), srate, signi,dc,timint,scale
            real*4    xstart,xend,tstart,tend, rtoff
            real*4    rdat(size,size), idat(size,size), ampl(size,size)
            character*40 info(5)
            logical   ok
c
            call dlsini()
            call fkini()
c
            ufldfn= 1
            itrace= 1
            redvel= 0.0
            rtoff = 0.0
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
            call fkdef(0)
1           call fkget(size)
            call fkdef(1)
c
            call addsuff(iname,'_fk',oname,length)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c open input file
c
            call optdat (ipath,iname,0,recsz,ihand,ierr)
            if(ierr.LT.0) then
               call report('segy','dataset','F',
     &  ' Cannot find all dataset files')
               call exit(1)
            endif
c
c read segy header
            call rhead(ihand,0,ierr)
            if (ierr.LT.0) then
               call report('segy','dataset','F',
     &   ' Unable to read reel header file')
               call exit(1)
            endif
c
            nsampl = rhinsm
            if (rhisr.NE.0) srate = 1.0E6/rhisr
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c find first trace
c
            do while (ufldfn.lt.ifirst)
              call rtdh(ihand,itrace,ierr)
              if (ierr.LT.0) then
                call report('segy','dataset','F',
     &   ' Unable to read trace header file')
```

```
            call exit(1)
           endif
           if (ierr.EQ.1) then
            call report('segy','dataset','F',
   &    ' shot point range doesn"t exist')
            call exit(1)
           endif
           itrace=itrace+1
          end do
          itrace=itrace-1
          xstart=wrange/1000.0
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c READ IN DATA TO REAL ARRAY
c
      do 100 while (.true.)
           if (ufldfn.GE.(ifirst+size-1)) goto 110
           itrace=itrace+1
           call razout('Reading trace : ',itrace)
           call flush(6)
c
c read in trace header to common block
            call rtdh(ihand,itrace,ierr)
            if (ierr.LT.0) then
             call report('segy','dataset','F',
   &    ' Unable to read trace header file')
             call exit(1)
            endif
c
c check for end of file
            if (ierr.GT.0) then
             call report('segy','dataset','I',
   &    ' End of file reached')
             goto 110
            endif
c
c check for correct sample no. and rate
           if (unsamp.NE.0) nsampl = unsamp
           if (wsr.NE.0) srate = 1.0E6/wsr
           if (xrtoff.NE.0) rtoff = real(xrtoff)
           if (xstart.EQ.0.0) xstart=wrange/1000.0
          xend=wrange/1000.0
c
c get reduction velocity if necessary
           if ((war).and.(xrvel.ne.0.0)) then
            redvel = real(xrvel)/1000.0
           endif
c
c read in data
           call rtrd (ihand,itrace,buffer,nsampl,ierr)
           if (ierr.LT.0) then
            call report('segy','dataset','F',
   &    ' Unable to read trace data file ')
            call exit(1)
           endif
c
c copy data to array rdat
          k=itrace-ifirst+1
          call azero(size,rdat(1,k))
          call aacopy(buffer,toff,toff+size,rdat(1,k),1,size)
          call mean(size,rdat(1,k),dc)
          call azero(size,idat(1,k))
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
  100     continue
```

```
 110     continue
c
c calculate start+end times
        tstart=(rtoff/1000.0)+toff/srate
        tend=(rtoff/1000.0)+(toff+size-1)/srate
c
c data info
        info(1)=''
        write(info(1),'(a14,f6.1,a2,f6.1,a3)'),
   &              'offset range  ',xstart,' -',xend,' km'
        info(2)=''
        write(info(2),'(a15,f6.2,a2,f6.2,a2)'),
   &              'time interval  ',tstart,' -',tend,' s'
        info(3)=''
        write(info(3),'(a16,f6.2,a3)'), 'sample rate     ',srate,' /s'
        info(4)=''
        if(war) then
          write(info(4),'(a15,f4.1,a5)'), 'data reduced at',redvel,' km/s'
        endif
c
c fill in any spaces at end of array
        if (k.lt.size) then
          do j=k+1,size
            call azero(size,rdat(1,j))
            call azero(size,idat(1,j))
          enddo
        endif
c
c taper edges (20 samples)
        call taper2d(size,k,rdat,20,20)
        if ((nsampl-toff).lt.size) then
          do j=1,k
            call taper_2(nsampl-toff,rdat(1,j),1,20)
          enddo
        endif
c
c reverse in x if necessary
        if (.not.reversed) then
          call reverse2d(rdat,size,size)
        endif
c
c do 2D fft and  reorder wavenumber information
        signi=1.0
        call fft2d(rdat,idat,size,size,signi)
        call kord2d(rdat,size,size)
        call kord2d(idat,size,size)
        timint=1.0/srate
c
c ask whether to plot unfiltered data
        call usrmes('plot unfiltered f-k data - ',1)
        call askok(6,5,ok)
        if (ok) then
c
c get amplitude information
        do j=1,size
        do i=1,size
          ampl(i,j)=sqrt(rdat(i,j)**2+idat(i,j)**2)
        enddo
        enddo
c
c transpose f and k axes
        call transpose(ampl,size)
c
c normalise
        call norm(ampl,size*(size/2+1),1.0,scale)
        info(5)=''
```

```
          write(info(5),'(a24,e10.3)'),'amplitude normalised to ',1.0/scale
c
c plot f-k data
c NB only plot data below nyquist frequency
          if (pltype.eq.2) then
            call fkplot(ampl,size,(size/2)+1,trsep,timint,
     &                  nclass,class,driver,title,info)
          elseif (pltype.eq.3) then
            call fkplot3d(ampl,ampl,size,(size/2)+1,trsep,timint,
     &                  nclass,class,driver,title,info)
          endif
c
          endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c apply velocity/frequency filter
c
          call fkfilter(rdat,idat,size,size,trsep,timint,f_filt,v_filt,
     &                        fflag,vflag,redvel)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c plot filtered data ?
c
          call usrmes('plot filtered f-k data - ',1)
          call askok(6,5,ok)
          if(ok) then
c
c get amplitude information
           do j=1,size
            do i=1,size
              ampl(i,j)=sqrt(rdat(i,j)**2+idat(i,j)**2)
            enddo
           enddo
c
c transpose f and k axes
           call transpose(ampl,size)
c
c normalise
          call norm(ampl,size*(size/2+1),1.0,scale)
          info(5)='
          write(info(5),'(a24,e10.3)'),'amplitude normalised to ',1.0/scale
c
           if (pltype.eq.2) then
            call fkplot(ampl,size,(size/2)+1,trsep,timint,
     &                  nclass,class,driver,title,info)
           elseif (pltype.eq.3) then
            call fkplot3d(ampl,ampl,size,(size/2)+1,trsep,timint,
     &                  nclass,class,driver,title,info)
           endif
c
          endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c write out filtered data to file
c
          call usrmes('write filtered data - ',1)
          call askok(6,5,ok)
          if(ok) then
c
c inverse transform
          call kord2d(rdat,size,size)
          call kord2d(idat,size,size)
          signi=-1.0
          call fft2d(rdat,idat,size,size,signi)
          if (.not.reversed) then
            call reverse2d(rdat,size,size)
```

```
          call reverse2d(idat,size,size)
         endif
c
          call optdat (ipath,oname,1,recsz/decifact,ohand,ierr)
          if(ierr.LT.0) then
           call report('segy','dataset','F',
     &   ' Cannot create output files - exist?')
           call exit(1)
          endif
c
          itrace=1
          otrace=1
          ufldfn=0
c
          do while (ufldfn.lt.ifirst)
           call rtdh(ihand,itrace,ierr)
           if (ierr.LT.0) then
            call report('segy','dataset','F',
     &    ' Unable to read trace header file')
            call exit(1)
           endif
           itrace=itrace+1
          end do
          itrace=itrace-1
c
      do 200 while (.true.)
           if (ufldfn.GE.(ifirst+size-1)) goto 210
           call razout('Writing trace : ',otrace)
           call flush(6)
c
           call rtdh(ihand,itrace,ierr)
           if (ierr.LT.0) then
            call report('segy','dataset','F',
     &    ' Unable to read trace header file')
            call exit(1)
           endif
c
           if (ierr.GT.0) then
            call report('segy','dataset','I',
     &    ' End of file reached')
            goto 210
           endif
c
c copy data to buffer
c nb pad traces with non-zero constant so agc in 'razzle' can cope
           call pad(buffer,1,16384,0.001)
           call aacopy(rdat(1,otrace),1,size,buffer,toff,toff+size)
c
           call wtrd (ohand,otrace,buffer,nsampl/decifact,ierr)
           if (ierr.LT.0) then
            call report('segy','dataset','F',
     &    ' Unable to write to trace data file ')
            call exit(1)
           endif
c
           unsamp=unsamp/decifact
           wsr=wsr*real(decifact)
c
           call wtdh(ohand,otrace,ierr)
           if (ierr.LT.0) then
            call report('segy','dataset','F',
     &    ' Unable to write to trace header file')
            call exit(1)
           endif
c
           itrace=itrace+1
```

```
            otrace=otrace+1
 200        continue
 210        continue
c
            rhinsm=rhinsm/decifact
            rhinsf=rhinsf/decifact
            rhisr =rhisr *decifact
            rhisrf=rhisrf*decifact
c
            call whead(ohand,0,ierr)
            if (ierr.LT.0) then
              call report('segy','dataset','F',
     &    ' Unable to write reel header file')
              call exit(1)
            endif
c
       call cltdat(ohand)
           endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      call cltdat(ihand)
      stop
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

            subroutine fkini()
c
c  menu default settings for f-k
c
            include "/home/kanga/dgl3dpg/fortran/include/f-k.h"
c
c character
            ipath = './ '
            iname = 'l1g128_8 '
            driver= 'mx11 '
            title=  'F-K spectrum'
            vflag = 'YES '
            fflag = 'NO  '
c logical
            reversed=.false.
            war    =.true.
c numerical
            ifirst= 1
            toff  = 600
            trsep = 75.0
            nclass= 14
            pltype= 2
c
c filter parameters
            data f_filt /  2.0,   4.0,       40.0,      50.0 /,
     &           v_filt /  4.0,    6.0,       8.0,       11.0 /
c
c class limits
            data class /0.005,            0.010,            0.05,
     &                  0.10,             0.15,             0.20,
     &                  0.25,             0.30,             0.35,
     &                  0.40,             0.45,             0.50,
     &                  0.55,             0.60,             0.65,
     &                  0.70,             0.75,             0.80,
     &                  0.85,             0.90 /
c
            return
            end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
      subroutine fkdef(mode)
c
c  mode = 0 -     read default values from file fk.dat if it exists
c                 else generate defaults
c           1 -   write menu values to file fk.dat
c
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dgl3dpg/fortran/include/f-k.h"
c
      integer*4 mode,iunit
      character*6 filename
c
      iunit=igetc()
      filename='fk.dat'
c
c check if file exists
      if (exists(filename)) then
c open as OLD file
       open(  unit=iunit, file=filename, status='old',
     &        form='formatted', iostat=ierr)
c
c else open as NEW file
      elseif(mode.ne.0) then
       open(  unit=iunit, file=filename, status='new',
     &        form='formatted', iostat=ierr)
      else
       call ifreec(iunit)
       return
      endif
c
      call fkrw(iunit,mode)
      close(unit=iunit)
      call ifreec(iunit)
c
      return
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      subroutine fkget(size)
c
c get parameter values for f-k from user via menu
c
      include "/home/kanga/dgl3dpg/fortran/include/f-k.h"
      integer*4 size
c
    do while (param.NE.99)
      call fkmenu(size)
      print*
      print 4
      read*,param
      print*
      if (param.EQ.1) then
            print*,'enter new input pathname :'
            read 5,ipath
      elseif (param.EQ.2) then
            print*,'enter new input filename :'
            read 5,iname
      elseif (param.EQ.3) then
            if(war) then
             war = .false.
            else
             war = .true.
            endif
      elseif (param.EQ.4) then
```

```
            print*,'enter first trace number :'
            read 8,ifirst
elseif (param.EQ.5) then
            print*,'enter first sample number :'
            read 8,toff
elseif (param.EQ.6) then
            print*,'enter trace separation :'
            read 7,trsep
elseif (param.EQ.7) then
            print*,'enter number of classes :'
            read 8,nclass
elseif (param.EQ.8) then
             call classdef(nclass,class)
elseif (param.EQ.9) then
            if(pltype.eq.2) then
             pltype=3
            else
             pltype=2
            endif
elseif (param.EQ.10) then
            print*,'enter device name :'
            read 5,driver
elseif (param.EQ.11) then
            if(vflag(1:1).eq.'Y'.or.vflag(1:1).eq.'y') then
             vflag='NO '
            elseif(vflag(1:1).eq.'N'.or.vflag(1:1).eq.'n') then
             print*,'enter velocity filter parameters :'
             read 9,v_filt
             vflag='YES '
            endif
elseif (param.EQ.12) then
            if(fflag(1:1).eq.'Y'.or.fflag(1:1).eq.'y') then
             fflag='NO '
            elseif(fflag(1:1).eq.'N'.or.fflag(1:1).eq.'n') then
             print*,'enter frequency filter parameters :'
             read 9,f_filt
             fflag='YES '
            endif
        elseif (param.EQ.13) then
            if (reversed) then
             reversed=.false.
            else
             reversed=.true.
            endif
        elseif (param.EQ.14) then
            print*,'enter title for plot : '
            read 5, title
        elseif (param.EQ.90) then
            call report('f-k transform','menu','I',
     &      'user requested abort')
            call exit(1)
        endif
      end do
c
   4    format (2X,'Enter number of parameter to alter (99 to continue) : ',$)
   5    format (A)
   7    format (F8.0)
   8    format (I4)
   9    format (4f5.0)
c
        return
        end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
      subroutine classdef(nclass,class)
c
      integer*4 nclass
      real*4    class(20),tclass
c
      if(nclass.ge.0) then
       iclass=1
       do while (iclass.ne.0)
        print 300
        do i=1,nclass
         print 301,i,class(i)
        enddo
        print 302
        read 303,iclass,tclass
        if(iclass.ge.1.and.iclass.le.nclass) then
         class(iclass)=tclass
        endif
       enddo
      else
       print 400
       call delay(5.)
      endif
c
 300  format(1x,'\f',4x,'class          value')
 301  format(1x,t7,i2,t15,f10.3)
 302  format(2x,'enter class number, value (return to exit) :')
 303  format(i,f11.0)
 400  format(1x,'\f',5x,'class boundaries calculated as percentiles')
c
      return
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
      subroutine fkplot(data,nk,nf,dx,dt,ncl,acl,
     &                  driver,title,info)
c
c adapted from UNIRAS AGL/CONTOURS example programs
c                  con007 + con 002
c
c nk,nf : grid dimensions
c ncl   : no. class limits (-ve = percentiles)
c ncol  : colour scale
c dx    : distance sample interval in m
c dt    : time sample interval in sec
c data(nk,nf):f-k data array
c
      parameter (ncol=1,iundef=9999)
      parameter (fundef=999.999)
c
      real*4    data(nk,nf), acl(20), dt,dx
      real*4    kmin,kmax,fmin,fmax,height,h
      integer*4 lenar1(4), lenar2(3), plotnum /1/
      character txtar1(4)*10, txtar2(3)*5
      character driver*(*),driver_str*30
      character title*(*),info(5)*(40),uni_str*12
c
      save plotnum
c user coord limits
      fmin=0.0
      fmax=1.0/(2.0*dt)
      kmin=-1.0/(2.0*dx)
      kmax= 1.0/(2.0*dx)
      call minmax(nk*nf,data,amin,amax)
c
c character string lengths of axis texts and axis text strings
      data lenar1 / 10,2,9,2 /
```

```
      data txtar1 / 'wavenumber','/m','frequency','Hz' /
c
c spacing between numeric labels on k and f axes
c there are 3 tickmarks between each label
      nticks = 4
      dblf=5.0
      dblk=0.5E-3
c
c character string lengths and character strings for colour scale
      data lenar2 /5,5,3/
      data txtar2 /'below','above','???'/
c
c open UNIRAS
      driver_str='sel "//driver//";exit'
      write(uni_str,'("UNIPICT.",I1)') plotnum
      plotnum=plotnum+1
      call gbegin (driver_str, ' ', uni_str)
c
c set limits and viewport
      call grpsiz(xsi,ysi)
      xm=min(xsi,ysi)
      xsize = 0.6*xm
      ysize = xsize
      xoff  = 0.25*(xsi-xsize)
      yoff  = 0.5*(ysi-ysize)
      call glimit(kmin,kmax,fmin,fmax,amin,amax)
      call gvport(xoff,yoff,xsize,ysize)
      call gwbox(1.0,1.0,0.0)
c
c set class levels
      if (ncl.gt.0) then
        call rclass(acl,ncl,0)
      else
c percentiles
        call rclass(0.0,-ncl,4)
      endif
c
c use colours from colour scale ncol
      call rshade(ncol,0)
c
c plot the 2d contour map (quick contour for large datasets)
      if (nk*nf.le.512*512) then
        call gcnr2s(data,nk,nf)
      else
        call grsz2q(0,0.5,0.5)
        call gcnr2q(data,nk,nf)
      endif
c
      height = 0.03*min(xsize,ysize)
c
c plot 4 axes
      call gscale
      call raxtef(4,'cenb',1)
      call raxsti(nticks)
      call raxdis(3,1,iundef)
c
c plot  k-axes
      call raxtex(6,lenar1(1),txtar1(1),fundef,fundef,height)
      call raxtex(5,lenar1(2),txtar1(2),fundef,fundef,0.8*height)
      call raxlfo(2,-3,iundef,iundef)
      call raxbti(iundef,fundef,fundef,dblk)
c bottom
      call raxdis(6,1,iundef)
      call raxdis(5,1,iundef)
      call raxis(1,fmin,height,1)
c top
```

```
      call raxdis(6,0,iundef)
      call raxdis(5,0,iundef)
      call raxdis(4,0,iundef)
      call raxis(1,fmax,height,2)
c
c plot  f-axes
      call raxtex(6,lenar1(3),txtar1(3),fundef,fundef,height)
      call raxtex(5,lenar1(4),txtar1(4),fundef,fundef,0.8*height)
      call raxlfo(0,0,iundef,0)
      call raxbti(iundef,fundef,fundef,dblf)
c left
      call raxdis(6,1,iundef)
      call raxdis(5,1,iundef)
      call raxdis(4,1,iundef)
      call raxis(2,kmin,height,1)
c  right
      call raxdis(6,0,iundef)
      call raxdis(5,0,iundef)
      call raxdis(4,0,iundef)
      call raxis(2,kmax,height,2)
c
c plot legend showing correspondence between colour & amplitude
      if (ncl.ge.15) then
        height=0.7*ysize/real(iabs(ncl))
        h=0.8*(fmax-fmin)/real(iabs(ncl))
      else
        height=0.7*ysize/15.0
        h=0.8*(fmax-fmin)/15.0
      endif
      call rtxfon('cenb',1)
      call gclopt(lenar2,txtar2,height,
     &            3,0.0,1)
      call gcoscl(kmax+0.1*(kmax-kmin),fmin)
c
c give plot a title
      tpf=fmax+0.2*(fmax-fmin)
      tpk=0.5*(kmin+kmax)
      call rtxjus(1,3)
      call rtxhei(5.0)
      call rtx(-1,title,tpk,tpf)
c
c add   additional info
      call rtxjus(0,0)
      call rtxhei(3.0)
      do i=1,5
       call rtx(-1,info(i),
     &  (kmax+0.05*(kmax-kmin)),(tpf-real(i)*h))
      enddo
c
c close UNIRAS
      call gend ()
      return
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      subroutine fkplot3d(ante3d,ante4d,nk,nf,dx,dt,ncl,
     &            acl,driver,title,info)
c
c adapted from UNIRAS AGL/CONTOURS example program con007
c
c np    : no. samples
c nk,nf : grid dimensions
c ncl   : no. class limits (-ve = percentiles)
c ncol  : colour scale
c dx    : distance sample interval in m
```

```
c dt        : time sample interval in sec
c ante3d(nk,nf) : data for 3rd dimension (z)
c ante4d(nk,nf) : data for 4th dimension (colour contours)
c
          parameter (ncol=1,iundef=9999)
          parameter (fundef=999.999)
c
          real*4    ante3d(nk,nf), ante4d(nk,nf), acl(20),dx,dt
          real*4      fmin,fmax,kmin,kmax
          integer*4 lenar1(6), lenar2(3), iplane(3), plotnum/1/
          character txtar1(6)*10, txtar2(3)*5
          character driver*(*),driver_str*30
          character title*(*),info(5)*40,uni_str*12
c
          save plotnum
c
c user coord limits
          fmin=0.0
          fmax=1.0/(2.0*dt)
          kmin=-1.0/(2.0*dx)
          kmax= 1.0/(2.0*dx)
c
c         call minmax(nk*nf,ante3d,zmin,zmax)
          zmin=0.0
          zmax=1.0
c
c contour line widths
          data wid / 0.3 /
c
c character string lengths of axis texts and axis text strings
          data lenar1 / -2, -2, -2, -2, -2, -2 /
          data txtar1 / '/m','wavenumber','Hz','frequency',
     &                  ' ','amplitude' /
c
c spacing between numeric labels on x, y and z axes
c there are 4 tickmarks between each label
          nticks = 4
          dblf=5.0
          dblk=0.5E-3
          dblz=0.25
c
c planes for axis annotation
          data iplane /1,1,1/
c
c character string lengths and character strings for colour scale
          data lenar2 /5,5,0/
          data txtar2 /'below ','above',' '/
c
c open UNIRAS
          driver_str='sel '//driver//';exit'
          write(uni_str,'("UNIPICT.",I1)') plotnum
          plotnum=plotnum+1
          call gbegin (driver_str, ' ', uni_str)
c
c set limits and viewport
          call grpsiz(xsi,ysi)
          xm=min(xsi,ysi)
          xsize = 0.8*xm
          ysize = xsize
          xoff  = 0.25*(xsi-xsize)
          yoff  = 0.25*(ysi-ysize)
          call glimit(kmin,kmax,fmin,fmax,zmin,zmax)
          call gvport(xoff,yoff,xsize,ysize)
c
c set 3d viewing parameters
          call gwbox(1.0,1.0,0.5)
```

```
      call geye(-3.0,-4.0,3.5)
c
c plot 3d axis system - x, y and z axes plotted separately because
c axis attributes differ
      call gvproj(2)
      height = 0.035*min(xsize,ysize)
c
c text for labels and axis texts
      call raxtef(4,'cenb',1)
      call raxtef(6,'cenb',1)
c
c turn plotting of minor tickmarks, axis texts and tick lines on,
c and set attributes to get integer type labels
      call raxdis(3,1,iundef)
      call raxdis(6,1,iundef)
      call raxdis(7,1,iundef)
c
c number of minor tickmarks
      call raxsti(nticks)
c
c plot the x
      call raxlfo(2,-3,iundef,iundef)
      call raxbti(iundef,fundef,fundef,dblk)
      call raxis3(1,height,iplane,lenar1,txtar1)
c
c plot the y axis
      call raxlfo(0,0,iundef,0)
      call raxbti(iundef,fundef,fundef,dblf)
      call raxis3(2,height,iplane,lenar1,txtar1)
c
c plot z axis
      call raxlfo(2,0,iundef,0)
      call raxbti(iundef,fundef,fundef,dblz)
      call raxis3(3,height,iplane,lenar1,txtar1)
c
c set class levels
      if (ncl.gt.0) then
       call rclass(acl,ncl,0)
      else
c percentiles
       call rclass(0.0,-ncl,4)
      endif
c
c use colours from colour scale ncol
      call rshade(ncol,0)
c
c define contour line widths
c when only one width defined - applies to all contour lines
      call gconwi(wid,1)
c
c set fishnet attributes
c every 5th gridline is plotted in anti-background colour
      call gconli(0,0.0,0,5.0,5.0,0.2,1)
c
c switch off the plotting of walls
      call gwall(9999)
c
c plot the 4d contour map
      call gconr4(ante3d,ante4d,nk,nf)
c
c plot legend showing correspondence between colour & signal strength
      if (ncl.ge.15) then
       height=0.55*ysize/real(iabs(ncl))
      else
       height=0.55*ysize/15.0
      endif
```

```
      call rtxfon('cenb',1)
      call gclopt(lenar2,txtar2,height,
     &            3,0.0,1)
      call gcoscl(xoff+1.15*xsize,yoff+0.05*ysize)
c
c give plot a title
      tpf=yoff+1.1*ysize
      tpk=xoff+0.6*xsize
      call rtxfon('swim',1)
      call rtxjus(1,3)
      call rtxhei(5.0)
      call rtx(-1,title,tpk,tpf)
c
c add   additional info
      call rtxfon('cenb',1)
      call rtxjus(0,0)
      call rtxhei(3.5)
      do i=1,5
       call rtx(-1,info(i),
     &  (xoff+0.9*xsize),(0.9*tpf-real(i)*5.0))
      enddo
c
c close UNIRAS
      call gend ()
      return
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      subroutine fkfilter(rdat,idat,nk,nf,dx,dt,f_filt,v_filt,
     &                    fflag,vflag,redvel)
c
c     Design and apply frequency and velocity (slowness) filters
c     with Hanning windows
c
c NB if fsize or psize = 0 then corresponding filter not applied
c
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      integer*4 nk, nf, fsize, psize, fs, ps
c fsize,psize = no. points defining half of window
      parameter(fsize=20,psize=20)
      real*4   rdat(nk,nf), idat(nk,nf), dx, dt
      real*4   fvals(16000), kvals(16000)
      real*4   f_filt(4), v_filt(4),rp_filt(4), redvel
      real*4   f(2*fsize), fa(2*fsize), p(2*psize), pa(2*psize)
      real*4    pi, arg
      parameter(pi=3.1415967)
      character*4 fflag,vflag
c
c calculate (reduced) slownesses
      do i=1,4
       if (redvel.eq.0.0) then
        rp_filt(5-i)=1.0/v_filt(i)
       else
        rp_filt(5-i)=1.0/v_filt(i)-1.0/redvel
       endif
      enddo
c
c calculate freq. ordinates (Hz)
      do j=1,nf/2
       a=real(j)
       b=real(nf)
       fvals(j)=(a-1.0)/(b*dt)
       fvals(nf-j+1)=-a/(b*dt)
```

```
        enddo
c
c calculate wavenumber ordinates in (1/km)
        do j=1,nk
         a=real(j)
         b=real(nk)
         kvals(j)=1000.0*((a-1.0)/(b*dx) - 0.5/dx)
        enddo
c
c check whether frequency filter is required
        if(fflag(1:1).eq.'Y'.or.fflag(1:1).eq.'y') then
         fs=fsize
        else
         fs=0
         goto 100
        endif
c
c design frequency filter window
        call azero(f,2*fsize)
        fa(1)=fvals(1)
        fa(2*fsize)=fvals(nf/2)
        do i=2,fsize
          fa(i)=f_filt(1)+
     &          real(i-2)*(f_filt(2)-f_filt(1))/real(fsize-2)
          arg=pi*real(i-2)/real(fsize-2)
          f(i)=0.5*(1.0-cos(arg))
c
          fa(i+fsize-1)=f_filt(3)+
     &          real(i-2)*(f_filt(4)-f_filt(3))/real(fsize-2)
          f(i+fsize-1)=0.5*(1.0+cos(arg))
        enddo
c
c check whether velocity filter is required
100     if(vflag(1:1).eq.'Y'.or.vflag(1:1).eq.'y') then
         ps=psize
        else
         ps=0
         goto 200
        endif
c
c design slowness filter window
        call azero(p,2*psize)
        pa(1)=kvals(1)/fvals(1)
        pa(2*psize)=kvals(nk)/fvals(1)
        do i=2,psize
          pa(i)=rp_filt(1)+
     &          real(i-2)*(rp_filt(2)-rp_filt(1))/real(psize-2)
          arg=pi*real(i-2)/real(psize-2)
          p(i)=0.5*(1.0-cos(arg))
c
          pa(i+psize-1)=rp_filt(3)+
     &          real(i-2)*(rp_filt(4)-rp_filt(3))/real(psize-2)
          p(i+psize-1)=0.5*(1.0+cos(arg))
        enddo
c
c apply filter
200     call fk_filt(fvals,kvals,rdat,idat,nf,nk,
     &          pa,p,2*ps,fa,f,2*fs)
c
        return
        end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
      subroutine fkrw(iunit,mode)
c
c read/write f-k defaults file fk.dat
c
      include "/home/kanga/dgl3dpg/fortran/include/f-k.h"
c
      integer*4 iunit, mode
c
c MODE=0 - read
      if (mode.eq.0) then
       read (iunit,100)  ifirst,   toff,   pltype,
     &                     nclass
       read (iunit,101)  trsep
       read (iunit,102)  class
       read (iunit,103)  v_filt,   f_filt
       read (iunit,104)  iname,    ipath,   driver,
     &                     title,    vflag,   fflag
       read (iunit,105)  reversed,  war
c
c MODE=1 - write
      elseif (mode.eq.1) then
       write (iunit,100)  ifirst,   toff,   pltype,
     &                     nclass
       write (iunit,101)  trsep
       write (iunit,102)  class
       write (iunit,103)  v_filt,   f_filt
       write (iunit,104)  iname,    ipath,   driver,
     &                     title,    vflag,   fflag
       write (iunit,105)  reversed,  war
c
      endif
c
100   format(4i4)
101   format(f7.2)
102   format(20f6.3)
103   format(8f5.2)
104   format(a20,a40,a15,a24,2a3)
105   format(2l1)
c
      return
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      subroutine fkmenu(size)
c
c     display f-k menu on screen
c
      include "/home/kanga/dgl3dpg/fortran/include/f-k.h"
      integer*4 size
      character*10 type
c
      type='???'
      if (pltype.eq.2) type='2-D'
      if (pltype.eq.3) type='3-D'
c
      print 199
      print 200,size,size
      print 201,ipath
      print 202,iname
      print 203,war
      print 204,ifirst
      print 205,toff
      print 206,trsep
      print 207,nclass
      print 208
```

```
      print 209,type
      print 210,driver
      print 211,v_filt,vflag
      print 212,f_filt,fflag
      print 213,reversed
      print 214,title
      print 290
      print 299
c
 199  format(1X,'\f\n',
     &       t18,'********************',/,
     &       t18,'*  F-K transform  *')
 200  format(t18,'*   ',i4,' x',i4,'    *',/,
     &       t18,'********************')
 201  format(2X,'1. Input pathname        ',T48,A32)
 202  format(2X,'2. Filename              ',T48,A32)
 203  format(2X,'3. Use WAR reduction info  ',T53,L1)
 204  format(2X,'4. First trace number     ',T50,I4)
 205  format(2X,'5. Time offset in samples  ',T50,I4)
 206  format(2X,'6. Trace separation       ',T50,F6.1)
 207  format(2X,'7. No. contour classes    ',T50,I4)
 208  format(2X,'8. Alter contour classes')
 209  format(2X,'9. Plot type (2-D or 3-D)  ',T52,A3)
 210  format(1X,'10. Device name               ',T46,A10)
 211  format(1X,'11. Velocity  filter(',4F5.1,' )',T52,A3)
 212  format(1X,'12. Frequency filter(',4F5.1,' )',T52,A3)
 213  format(1X,'13. Reversed data',T53,L1)
 214  format(1X,'14. Title                ',T48,A24)
 290  format(1X,'\n',1X,'90.                    ABORT')
 299  format(1X,'\n',1X,'99.             RUN ')
c
      return
      end
```



```
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      subroutine fk_filt(xf,xk,x,y,lxf,lxk,pa,p,np,fa,f,nf)
c
c slowness-frequency filtering in the f-k domain
c
c dls 87c
c adapted - dpg 91
c
c  x(lxf,lxk)  2d data array  of real part
c  y(lxf,lxk)  2d data array of imaginary part
c  xf(lxf) ,xk(lxk) arrays containing ordinates
c  p(np),pa(np) slowness filter values and ordinates
c         if np=0 then no filter is applied
c  f(nf),fa(nf) angular frequency filter values and ordinates
c         if nf=0 then the nyquist is assumed
c
c  the filters are interpolated linearly to obtain coefficients
c  at the requred velocity and frequency values
c  frequency filters are assumed symmetric about f=0.0
c
      parameter(maxlen=2048)
      real*4 x(lxf,lxk),xf(lxf),xk(lxk),p(np),pa(np),f(nf),fa(nf)
      real*4 y(lxf,lxk)
      real*4 ffilt(maxlen), slowness
c
c store frequency filter coefficients to save time
      if(nf.gt.0) then
      do 100 jf=1,lxf
        call linint(fa,f,nf,abs(xf(jf)),ffilt(jf))
 100  continue
      else
```

```
      call pad(ffilt,1,lxf,1.0)
    endif
c
c now execute the filter
    do 300 jk=1,lxk
      do 200 jf=1,lxf
c
c calc velocity filter
        if(np.gt.0.and.xf(jf).ne.0.0) then
          slowness=xk(jk)/xf(jf)
          call linint(pa,p,np,slowness,pfcoff)
        else
          pfcoff=1.0
        endif
c
c apply filter
        x(jf,jk)=x(jf,jk)*ffilt(jf)*pfcoff
        y(jf,jk)=y(jf,jk)*ffilt(jf)*pfcoff
c
200     continue
300   continue
    return
    end
```

5. F-T

```
      program f_t
c
c accepts trace file sequence and outputs frequency amplitude
c spectrum trace sequence
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      integer*4 ierr, ihand, ohand, recsz, new_recsz
      integer*4 param
      integer*4 itrace, otrace, ifirst, ilast, nrec
      integer*4 len,l1,l2,nsampl
      real*4    work(16384),x(16384),y(16384),srate,df
      character iname*20, oname*24, ipath*25, opath*25
c
      call dlsini()
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c defaults
c
      ipath = './                '
      opath = './                '
      iname = '1_l60bs5          '
      oname = '                  '
c
c ifirst is trace no. at start of loop, ilast is
c trace no. at end, and nrec is step size
      ifirst = 1
      ilast  = 9999
      nrec   = 1
      otrace = 0
c

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c get parameter values from user via menu
c
  1   call menu(ipath,opath,iname,ifirst,ilast,nrec)
      print*
      print 4
      read*,param
      print*
      if (param.EQ.99) goto 9
      if (param.EQ.1) then
            print*,'enter new input pathname :'
            read 5,ipath
      elseif (param.EQ.2) then
            print*,'enter new output pathname :'
            read 5,opath
      elseif (param.EQ.3) then
            print*,'enter new filename :'
            read 5,iname
      elseif (param.EQ.4) then
            print*,'enter first trace number :'
            read 6,ifirst
      elseif (param.EQ.5) then
            print*,'enter last trace number :'
            read 6,ilast
      elseif (param.EQ.6) then
            print*,'enter trace step :'
            read 6,nrec
      endif
      go to 1
c
  4   format (2X,'Enter number of parameter to alter (99 to continue) : ',$)
```

```
  5      format (A)
  6      format (I4)
c
  9      continue
c
         oname = iname(1:lnb(iname))//'_ft'
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c open input filesc
         call optdat (ipath,iname,0,recsz,ihand,ierr)
         if(ierr.LT.0) then
           call report('segy','dataset','F',
   &  ' Cannot find all dataset files')
           call exit(1)
         endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c read segy header
c
         call rhead(ihand,0,ierr)
         if (ierr.LT.0) then
           call report('segy','dataset','F',
   &   ' Unable to read reel header file')
           call exit(1)
         endif
c
         nsampl = rhinsm
         if (rhisr.NE.0) srate = 1.0E6/rhisr
c
    len = nsampl
    do 90 i=1,13
      if(nsampl.gt.(2**i).and.nsampl.le.(2**(i+1)) ) then
        len=2**(i+1)
      endif
 90  continue
c
    l1 = len / 2 + 1
    l2 = l1 + 1
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c open output files
c
         new_recsz=l1*4
     call optdat (opath,oname,1,new_recsz,ohand,ierr)
c
         if(ierr.LT.0) then
           call report('segy','dataset','F',
   &   ' Unable to create dataset - space? or exists?')
           call exit(1)
         endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
    do 10 itrace=ifirst,ilast,nrec
         call razout('Transforming trace ', itrace)
         call flush(6)
         otrace=otrace+1
c
c read in trace header to common block
c
         call rtdh(ihand,itrace,ierr)
         if (ierr.LT.0) then
           call report('segy','dataset','F',
   &   ' Unable to read trace header file')
           call exit(1)
         endif
```

```
c
c check for end of file
            if (ierr.GT.0) then
              call report('segy','dataset','I',
     &     ' End of file reached')
              goto 11
            endif
c
            if (unsamp.NE.0) nsampl = unsamp
            if (wsr.NE.0) srate = 1.0E6/wsr
c
c read in data + write to new file
            call rtrd (ihand,itrace,work,nsampl,ierr)
            if (ierr.LT.0) then
              call report('segy','dataset','F',
     &     ' Unable to read trace data file ')
              call exit(1)
            endif
c
c cosine taper both ends of array to zero over 0.5 seconds
            call taper(nsampl,work,int(0.5*srate))
c
c copy to working array
        call azero(16384,x)
            call aacopy(work,1,nsampl,x,1,nsampl)
c
c   transform the time series
        call azero(len,y)
        signi = -1.0
        call ffta(len, x, y, signi)
c
c get amplitude
            do j=1,len
              work(j)=sqrt(x(j)**2+y(j)**2)
            enddo
c
c calc the frequency interval
        df = srate / dfloat(len)
c
c write out 1st half of frequency series
            call wtrd (ohand,otrace,work,l1,ierr)
            if (ierr.LT.0) then
              call report('segy','trace data','F',
     &     ' Unable to write to file')
              call cltdat(ohand)
              call exit(1)
            endif
c
c amend trace header
            ufldfn=otrace
            ufldtn=ufldfn
            unsamp=l1
            wsr=df*1.0E6
c
c fill in some other variables required by plot program
            if (ufldfn.EQ.0) ufldfn = ufldtn
            if (utrace.EQ.0) utrace = 1
            ushot =0
c
            call wtdh(ohand,otrace,ierr)
            if (ierr.LT.0) then
              call report('segy','trace data','F',
     &     ' Unable to write to file')
              call cltdat(ohand)
              call exit(1)
            endif
```

```
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
 10  continue
 11  continue
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      print 100,ushot ,utrc  ,ufldfn,ufldtn,usrcpn,ucdp,
     &      utrace,utrtyp,unsum ,unstk ,uduse ,
     &      wrange,wrcvge,wsrcse,wsrcdp,wrcvde,wsrcde,
     &                wsrcwd,wrcvwd,
     &                wsrcx ,wsrcy ,wrcvx ,wrcvy ,ucunit,
     &                wvelw ,wvelsw,usrcut,urcvut,usrcst,urcvst,
     &                utotst,ulaga ,ulagb ,udelt ,umutst,umutfn,
     &                unsamp,wsr   ,ugntyp,wgain ,wgnst ,
     &                ucorr ,ufrqst,ufrqfn,uslen ,ustyp ,ustlen,
     &                uetlen,utptyp,ualfrq,ualslp,unofrq,unoslp,
     &                ulofrq,uhifrq,uloslp,uhislp,uyear ,uday  ,
     &                uhour ,umin  ,usec  ,utctyp,xmsec ,
     &                xeyear ,xeday  ,xehour ,xemin  ,xesec  ,xemsec ,
     &      xdelt ,xrvel,xrtoff ,xstncd,xevecd
c
 100  format(6(I10),/,5(I10),/,6(F10.2),/,2(F10.2),/,4(F10.2),I10,/,
     &      2(F10.2),4(I10),/,6(I10),/,I10,F10.2,I10,2(F10.2),/,
     &      3(6(I10),/),5(I10),/,6(I10),/,3(I10),2(6X,A4),/)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c amend and write out new file header
c
          rhisr= int(df*1.0E6)
          rhisrf=rhisr
          rhinsm=l1
          rhinsf=rhinsm
c
      call whead(ohand,0,ierr)
      if (ierr.LT.0) then
            call report('segy','dataset','F',
     &    ' Unable to write reel header file')
            call exit(1)
      endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c dump common block rhicom + extras
c
          print*,'\n Rhicom common block'
          print 101,rhintr,rhinax,rhisr ,rhisrf,
     &          rhinsm,rhinsf,ifmt  ,rhirli,rhilni, rhimes
  101  format(1X,3(I8),/,1X,6(I8))
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      call cltdat(ihand)
      call cltdat(ohand)
c
      stop
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
          subroutine menu(pathi,patho,name,ifirst,ilast,nrec)
c
          character*(*) pathi, patho, name
          integer  ifirst, ilast, nrec
c
          print 200
```

```
      print 201,pathi
      print 202,patho
      print 203,name
      print 204,ifirst
      print 205,ilast
      print 206,nrec
      print 299
c
 200  format(10X,'**********************',/,
  &    10X,'* Frequency Transform *',/,
  &    10X,'**********************')
 201  format(2X,'1. Input pathname  :',T55,A25)
 202  format(2X,'2. Output pathname :',T55,A25)
 203  format(2X,'3. Filename        :',T55,A20)
 204  format(2X,'4. First trace number    :', T55,I4)
 205  format(2X,'5. Last trace number     :',T55,I4)
 206  format(2X,'6. Step                  :',T55,I4)
 299  format(1X,'99.            Run ')
c
      return
      end
```

6. NORMALISE

```
      program normalise
c
c  accepts trace file sequence and normalises data to +/- lim
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      integer*4 ierr, ihand, recsz, nsampl
      integer*4 param
      integer*4 itrace, ifirst, ilast
      character name*25, ipath*25
           real*4   intrace(16000),srate, lim,scale
c
      call dlsini()
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c defaults
c
      ipath = './                '
      name = '1_l6bs2            '
      lim=100.0
c
c ifirst is trace no. at start of loop, ilast is trace no. at end
      ifirst = 1
      ilast  = 9999
      itrace = 1
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c get parameter values from user via menu
c
      do while (param.ne.99)
            call menu(ipath,name,ifirst,ilast,lim)
            print*
            print 4
            read*,param
            print*
            if (param.EQ.1) then
                  print*,'enter new  pathname :'
                  read 5,ipath
            elseif (param.EQ.2) then
                  print*,'enter new filename :'
                  read 5,name
            elseif (param.EQ.3) then
                  print*,'enter first trace number :'
                  read 8,ifirst
            elseif (param.EQ.4) then
                  print*,'enter last trace number :'
                  read 8,ilast
            elseif (param.EQ.5) then
                  print*,'enter limit :'
                  read 6,lim
            elseif (param.EQ.90) then
                  call report('normalisation','menu','I',
     &            'user requested abort')
                  call exit(1)
            endif
      enddo
c
    4     format (2X,'Enter number of parameter to alter (99 to continue) : ',$)
    5     format (A)
    6     format (F5.2)
    8     format (I4)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
c open  files and read segy header
c
          call optdat (ipath,name,0,recsz,ihand,ierr)
c
          if(ierr.LT.0) then
            call report('normalise','dataset','F',
     &  ' Cannot find all dataset files')
            call exit(1)
          endif
c
          call rhead(ihand,0,ierr)
c
          if (ierr.LT.0) then
            call report('normalise','dataset','F',
     &    ' Unable to read reel header file')
            call exit(1)
          endif
c
          nsampl = rhinsm
          if (rhisr.NE.0) srate  = 1.0E6/rhisr
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c find first trace
c
          do while (ufldfn.lt.ifirst)
            call rtdh(ihand,itrace,ierr)
            if (ierr.LT.0) then
              call report('normalise','dataset','F',
     &    ' Unable to read trace header file')
              call exit(1)
            endif
            if (ierr.EQ.1) then
              call report('normalise','dataset','F',
     &    ' shot point range doesn"t exist')
              call exit(1)
            endif
            itrace=itrace+1
c
            call razclk()
            call flush(6)
c
          end do
          itrace=itrace-2
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c MAIN LOOP
c
      do 10 while (.true.)
            if (ufldfn.GE.ilast) goto 11
            itrace=itrace+1
c           print 1000,itrace
            call razout('Trace : ',itrace)
            call flush(6)
c
c read in trace header to common block
            call rtdh(ihand,itrace,ierr)
            if (ierr.LT.0) then
              call report('normalise','dataset','F',
     &    ' Unable to read trace header file')
              call exit(1)
            endif
c
c check for end of file
            if (ierr.GT.0) then
              call report('normalise','dataset','I',
     &    ' End of file reached')
```

```
            goto 11
           endif
c
c check for correct sample no. and rate
           if (unsamp.NE.0) nsampl = unsamp
           if (wsr.NE.0) srate = 1.0E6/wsr
c
c read in data
           call rtrd (ihand,itrace,intrace,nsampl,ierr)
c
           if (ierr.LT.0) then
            call report('normalise','dataset','F',
     &    ' Unable to read trace data file ')
            call exit(1)
           endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c remove dc
          call norm(intrace,nsampl,lim,scale)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c write to file
c
           call wtdh(ihand,itrace,ierr)
           if (ierr.LT.0) then
            call report('statics','dataset','F',
     &    ' Unable to write to trace header file')
            call exit(1)
           endif
c
           call wtrd (ihand,itrace,intrace,nsampl,ierr)
           if (ierr.LT.0) then
            call report('segy','trace data','F',
     &    ' Unable to write to file')
            call cltdat(ihand)
            call exit(1)
           endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
  10  continue
  11  continue
      call cltdat(ihand)
c
1000  format(1X,'Trace : ',I5)
c
      stop
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
          subroutine menu(pathi,name,ifirst,ilast,lim)
c
          character*(*) pathi, name
          integer  ifirst, ilast
c
          print*,'\f'
          print 200
          print 201,pathi
          print 202,name
          print 203,ifirst
          print 204,ilast
          print 205,lim
          print 290
          print 299
          print 300
```

```
c
 200    format(9X,'*****************',/,
  &      9X,'* NORMALISATION *',/,
  &      9X,'*****************')
 201    format(2X,'1. Input pathname  :',T45,A25)
 202    format(2X,'2. Filename        :',T45,A20)
 203    format(2X,'3. First trace number          :', T45,I4)
 204    format(2X,'4. Last trace number           :',T45,I4)
 205    format(2X,'5. Maximum data value required   :',T45,F5.1)
 290    format(1X,'\n',1X,'90             ABORT')
 299    format(1X,'\n',1X,'99.             RUN ')
c
 300    format(1X,'\n',1X,'Warning: original data will be overwritten')
c
        return
        end
```

7. PANEL

```
      program panel
c
c  accepts trace file sequence and band pass filters panel of traces
c  using a range of filter windows
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      integer*4 ierr, ihand, ohand, recsz
      integer*4 panel
      integer*4 i,itrace, newtrace, ifirst, panlen, nrec
      integer*4 iwopt,nsampl
      real*4    work(16384),srate,f1,f2,f3,f4
      real*4    lowf,hif,wlen,gap,slpl,slph
      character iname*20, oname*24, ipath*25, opath*25
c
      call dlsini()
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c defaults
            ipath  = './                '
            opath  = './                '
            iname = '1_110101           '
c
c window is hanning
            iwopt = 2
c
            lowf= 2.0
            hif = 25.0
            wlen= 5.0
            gap = 1.0
            slp = 2.0

c
c ifirst is trace no. at start of loop, panlen is
c no. traces in panel, and nrec is step size
            ifirst = 1
            panlen  = 18
            nrec   = 1
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c get parameter values from user via menu
      do while (param.NE.99)
            call menu(ipath,opath,iname,ifirst,lowf,hif,wlen,gap)
            print*
            print 4
            read*,param
            print*
            if (param.EQ.1) then
                  print*,'enter new input pathname :'
                  read 5,ipath
            elseif (param.EQ.2) then
                  print*,'enter new output pathname :'
                  read 5,opath
           elseif (param.EQ.3) then
                  print*,'enter new filename :'
                  read 5,iname
            elseif (param.EQ.4) then
                  print*,'enter first trace number :'
                  read 6,ifirst
            elseif (param.EQ.5) then
                  print*,'enter low frequency limit (>1.0) :'
```

```
                read 7,lowf
                if(lowf.le.1.0) then
                  print*,'must be > 1.0 :'
                read 7,lowf
                endif
        elseif (param.EQ.6) then
                print*,'enter high frequency limit :'
                read 7,hif
        elseif (param.EQ.7) then
                print*,'enter frequency window length :'
                read 7,wlen
        elseif (param.EQ.8) then
                print*,'enter interval between start of successive windows:'
                read 7,gap
        endif
    enddo
c
  4     format (2X,'Enter number of parameter to alter (99 to continue) : ')
  5     format (A)
  6     format (I)
  7     format (F5.1)
c
        oname = iname(1:lnb(iname))//'_pan'
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c open input files, read segy header and open output files
        call optdat (ipath,iname,0,recsz,ihand,ierr)
        if(ierr.LT.0) then
          call report('segy','dataset','F',
   &  ' Cannot find all dataset files')
          call exit(1)
        endif
c
        call rhead(ihand,0,ierr)
        if (ierr.LT.0) then
          call report('segy','dataset','F',
   &      ' Unable to read reel header file')
          call exit(1)
        endif
c
        nsampl = rhinsm
        if (rhisr.NE.0) srate  = 1.0E6/rhisr
c
        call optdat (opath,oname,1,recsz,ohand,ierr)
        if(ierr.LT.0) then
                call report('segy','dataset','F',
   &            ' Unable to create dataset - space? or exists?')
                call exit(1)
        endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
    newtrace = 0
    do 20 panel = 1,int((hif-wlen-lowf)/gap+1)
c
        freq=lowf+(panel-1)*gap
        slpl = sqrt(freq)
        slph = sqrt(freq+wlen)
c
        f1=freq-slpl
        f2=freq
        f3=freq+wlen
        f4=freq+wlen+slph
        if (f2.GT.f3) then
          f2 = (f2+f3)/2
```

```
          f3 = f2
         endif
c
         print 1000,panel,f1,f2,f3,f4
c
     do 10 i=1,panlen,nrec
          itrace = i + ifirst - 1
          newtrace = newtrace+1
          print 1001,itrace,newtrace
c
c read in trace header to common block
          call rtdh(ihand,itrace,ierr)
          if (ierr.LT.0) then
           call report('segy','dataset','F',
   &   ' Unable to read trace header file')
           call exit(1)
          endif
c
          if (nsampl.EQ.0) nsampl = unsamp
          if (srate.EQ.0.AND.wsr.NE.0) srate = 1.0E6/wsr
c
c read in data, apply filter + write to new file
          call rtrd (ihand,itrace,work,nsampl,ierr)
          if (ierr.LT.0) then
           call report('segy','dataset','F',
   &   ' Unable to read trace data file ')
           call exit(1)
          endif
c
          call bpass(nsampl,work,f1,f2,f3,f4,srate,iwopt)
c
          call wtrd (ohand,newtrace,work,nsampl,ierr)
          if (ierr.LT.0) then
           call report('segy','trace data','F',
   &   ' Unable to write to file')
           call cltdat(ohand)
           call exit(1)
          endif
c
c put filter values to common block variables and write to new header file
          ulofrq = int(f1)
          uhifrq = int(f4)
c
          ufldtn = newtrace
          ufldfn = newtrace
c
          call wtdh(ohand,newtrace,ierr)
          if (ierr.LT.0) then
           call report('segy','trace data','F',
   &   ' Unable to write to file')
           call cltdat(ohand)
           call exit(1)
          endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
  10  continue
c
c put in a couple of blank traces
     do 16 j=1,2
          newtrace=newtrace+1
          ufldtn = newtrace
          ufldfn = newtrace
          utrtyp = 3
      call wtrd(ohand,newtrace,work,nsample,ierr)
          call wtdh(ohand,newtrace,ierr)
```

```
          print 1002,newtrace
   16  continue
   20  continue
c
          print*,"total number of traces = ",newtrace
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c write out file header with extended name
          rhintr = newtrace
c
      call whead(ohand,0,ierr)
      if (ierr.LT.0) then
            call report('segy','dataset','F',
     &    ' Unable to write reel header file')
            call exit(1)
      endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      call cltdat(ihand)
      call cltdat(ohand)
c
 1000  format(1X,"Panel ",I3,5X,"low-cut =",2F6.1,5X,"high-cut =",2F6.1)
 1001  format(1X,"Filtering trace : ",I5,"->",I5)
 1002  format(1X,"Blank trace     : ",5X "->",I5)
c
      stop
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
          subroutine menu(pathi,patho,name,ifirst,lowf,hif,wlen,gap)
c
          character*(*) pathi, patho, name
          integer*4 ifirst
          real*4    lowf,hif,wlen,gap
c
          print 200
          print 201,pathi
          print 202,patho
          print 203,name
          print 204,ifirst
          print 205,lowf
          print 206,hif
          print 207,wlen
          print 208,gap
          print 209
c
   200    format(10X,'*************************',/,
     &      10X,'* BAND PASS FILTER PANELS*',/,
     &      10X,'*************************')
   201    format(2X,'1. Input pathname                :',T55,A25)
   202    format(2X,'2. Output pathname               :',T55,A25)
   203    format(2X,'3. Filename                    :',T55,A20)
   204    format(2X,'4. First trace number            :',T55,I5)
   205    format(2X,'5. Low  frequency limit          :',T55,F5.1)
   206    format(2X,'6. High frequency limit          :',T55,F5.1)
   207    format(2X,'7. Frequency window length         :',T55,F5.1)
   208    format(2X,'8. Interval between successive windows :',T55,F5.1)
   209    format(2X,'9.          RUN')
c
          return
          end
```

8. RMDC

```
      program rmdc
c
c  accepts trace file sequence and removes dc component
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      integer*4 ierr, ihand, recsz, nsampl
      integer*4 param
      integer*4 itrace, ifirst, ilast
      character name*25, ipath*25
            real*4   intrace(16000),srate, dc
c
      call dlsini()
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c defaults
c
      ipath = './                '
      name = '1_l6bs2            '
c
c ifirst is trace no. at start of loop, ilast is
c trace no. at end
      ifirst = 1
      ilast  = 9999
      itrace = 1
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c get parameter values from user via menu
c
      do while (param.ne.99)
            call menu(ipath,name,ifirst,ilast)
            print*
            print 4
            read*,param
            print*
            if (param.EQ.1) then
                  print*,'enter new  pathname :'
                  read 5,ipath
            elseif (param.EQ.2) then
                  print*,'enter new filename :'
                  read 5,name
            elseif (param.EQ.3) then
                  print*,'enter first trace number :'
                  read 8,ifirst
            elseif (param.EQ.4) then
                  print*,'enter last trace number :'
                  read 8,ilast
            elseif (param.EQ.90) then
                  call report('dc correction','menu','I',
     &            'user requested abort')
                  call exit(1)
            endif
      enddo
c
    4     format (2X,'Enter number of parameter to alter (99 to continue) : ',$)
    5     format (A)
    6     format (F5.2)
    8     format (I4)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c open  files and read segy header
c
            call optdat (ipath,name,0,recsz,ihand,ierr)
```

```
          if(ierr.LT.0) then
            call report('rmdc','dataset','F',
     &  ' Cannot find all dataset files')
            call exit(1)
          endif
c
          call rhead(ihand,0,ierr)
          if (ierr.LT.0) then
            call report('rmdc','dataset','F',
     &   ' Unable to read reel header file')
            call exit(1)
          endif
c
          nsampl = rhinsm
          if (rhisr.NE.0) srate  = 1.0E6/rhisr
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c find first trace
c
          do while (ufldfn.lt.ifirst)
           call rtdh(ihand,itrace,ierr)
           if (ierr.LT.0) then
            call report('rmdc','dataset','F',
     &   ' Unable to read trace header file')
            call exit(1)
           endif
           if (ierr.EQ.1) then
            call report('rmdc','dataset','F',
     &   ' shot point range doesn"t exist')
            call exit(1)
           endif
           itrace=itrace+1
c
           call razclk()
           call flush(6)
c
          end do
          itrace=itrace-2
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c MAIN LOOP
c
     do 10 while (.true.)
           if (ufldfn.GE.ilast) goto 11
           itrace=itrace+1
           call razout('Trace : ',itrace)
           call flush(6)
c
c read in trace header to common block
           call rtdh(ihand,itrace,ierr)
           if (ierr.LT.0) then
            call report('rmdc','dataset','F',
     &   ' Unable to read trace header file')
            call exit(1)
           endif
c
c check for end of file
           if (ierr.GT.0) then
            call report('rmdc','dataset','I',
     &   ' End of file reached')
            goto 11
           endif
c
c check for correct sample no. and rate **
           if (unsamp.NE.0) nsampl = unsamp
           if (wsr.NE.0) srate = 1.0E6/wsr
```

```
c
c read in data
          call rtrd (ihand,itrace,intrace,nsampl,ierr)
          if (ierr.LT.0) then
           call report('rmdc','dataset','F',
   &    ' Unable to read trace data file ')
           call exit(1)
          endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c remove dc
         call mean(nsampl,intrace,dc)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c write to file
c
          call wtdh(ihand,itrace,ierr)
          if (ierr.LT.0) then
           call report('statics','dataset','F',
   &    ' Unable to write to trace header file')
            call exit(1)
          endif
c
          call wtrd (ihand,itrace,intrace,nsampl,ierr)
          if (ierr.LT.0) then
           call report('segy','trace data','F',
   &    ' Unable to write to file')
            call cltdat(ihand)
            call exit(1)
          endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
```



```
 10  continue
 11  continue
     call cltdat(ihand)
c
1000  format(1X,'Trace : ',I5)
c
     stop
     end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
          subroutine menu(pathi,name,ifirst,ilast)
c
          character*(*) pathi, name
          integer  ifirst, ilast
c
          print*,'\f'
          print 200
          print 201,pathi
          print 202,name
          print 203,ifirst
          print 204,ilast
          print 290
          print 299
          print 300
c
  200     format(9X,'*****************',/,
    &       9X,'*D.C. CORRECTION*',/,
    &       9X,'*****************')
  201     format(2X,'1. Input pathname  :',T45,A25)
  202     format(2X,'2. Filename        :',T45,A20)
  203     format(2X,'3. First trace number          :', T45,I4)
  204     format(2X,'4. Last trace number           :',T45,I4)
```

```
 290    format(1X,'\n',1X,'90          ABORT')
 299    format(1X,'\n',1X,'99.         RUN ')
c
 300    format(1X,'\n',1X,'Warning: original data will be overwritten')
c
        return
        end
```

9. STACK

STACK.H

```
c
c       common block for variables in stacking program
c
        integer*4 maxstack
        parameter(maxstack=11)
c
        integer*4 stackpos(maxstack), id(maxstack)
c
        real*4    stackdat(16000,maxstack), x(maxstack),
     &            array(maxstack)
c
        common /stackcom/
     &          stackpos,id,stackdat,x,array
        save /stackcom/
c
```

STACK_MENU.H

```
c
c       common block for menu variables in stacking program
c
        integer*4 param
        integer*4 ifirst, ilast, channel
        integer*4 stacksiz, trorig
c
        real*4    stackvel, trsep, xorig
c
        character iname*20, oname*24, ipath*25, opath*25
        character war*3, flag*3
c
        common /menucom/
     &          param, ifirst, ilast, channel, stacksiz, trorig,
     &          stackvel, trsep, xorig, iname, oname, ipath, opath,
     &          war, flag
        save /menucom/
c
```

```
      program stack
c
c accepts trace file sequence and stacks with linear moveout
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      include "../include/stack_menu.h"
      include "../include/stack.h"
c
            integer*4 ihand, ohand,ierr, recsz
            integer*4 itrace,otrace, nsampl
            real*4    trace(16000), srate
c
       call dlsini()
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
            ufldfn= 1
            itrace= 1
            otrace= 1
c
            call menu_default()
            call menu()
c
c initialise stack parameters
            call stackini(stacksiz)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c open input and output files
c
            call optdat (ipath,iname,0,recsz,ihand,ierr)
c
            if(ierr.LT.0) then
               call report('segy','dataset','F',
     &  ' Cannot find all dataset files')
               call exit(1)
            endif
c
       call optdat (opath,oname,1,recsz,ohand,ierr)
c
            if(ierr.LT.0) then
               call report('segy','dataset','F',
     &   ' Unable to create dataset - space? or exists?')
               call exit(1)
            endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c read segy header
            call rhead(ihand,0,ierr)
c
            if (ierr.LT.0) then
               call report('segy','dataset','F',
     &   ' Unable to read reel header file')
               call exit(1)
            endif
c
            nsampl = rhinsm
            if (rhisr.NE.0) srate = 1.0E6/rhisr
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c find first trace
            do while (ufldfn.lt.ifirst)
```

```
          call rtdh(ihand,itrace,ierr)
          if (ierr.LT.0) then
           call report('segy','dataset','F',
     &   ' Unable to read trace header file')
           call exit(1)
          endif
          if (ierr.EQ.1) then
           call report('segy','dataset','F',
     &   ' shot point range doesn''t exist')
           call exit(1)
          endif
          itrace=itrace+1
         end do
         itrace=itrace-1
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c MAIN LOOP
c
     do 100 while (.true.)
          if (ufldfn.GE.ilast) goto 110
          itrace=itrace+1
          print 1000,itrace,otrace
c
c read in trace header to common block
           call rtdh(ihand,itrace,ierr)
           if (ierr.LT.0) then
            call report('segy','dataset','F',
     &   ' Unable to read trace header file')
            call exit(1)
           endif
c
c check for end of file
           if (ierr.GT.0) then
            call report('segy','dataset','I',
     &   ' End of file reached')
            goto 110
           endif
c
c check for correct channel no.,sample no. and rate
           if ((channel.NE.0).AND.(channel.NE.utrace)) goto 100
           if (nsampl.EQ.0) nsampl = unsamp
           if (srate.EQ.0.AND.wsr.NE.0) srate = 1.0E6/wsr
c
c read in data
           call rtrd (ihand,itrace,trace,nsampl,ierr)
c
           if (ierr.LT.0) then
            call report('segy','dataset','F',
     &   ' Unable to read trace data file ')
            call exit(1)
           endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
          call stk(trace,srate,nsampl)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c write to new file
           call wtrd (ohand,otrace,trace,nsampl,ierr)
c
           if (ierr.LT.0) then
            call report('segy','trace data','F',
     &   ' Unable to write to file')
            call cltdat(ohand)
            call exit(1)
```

```
            endif
c
            call wtdh(ohand,otrace,ierr)
            if (ierr.LT.0) then
              call report('segy','trace data','F',
     &    ' Unable to write to file')
              call cltdat(ohand)
              call exit(1)
            endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
          otrace=otrace+1
c
  100     continue
  110     continue
c
 1000   format(1X,' Trace : ',I5,' -> ',I5)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c write out file header with extended name
c
      call whead(ohand,0,ierr)
      if (ierr.LT.0) then
            call report('segy','dataset','F',
     &    ' Unable to write reel header file')
            call exit(1)
      endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      call cltdat(ihand)
      call cltdat(ohand)
c
      stop
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
          subroutine menu()
c
c get parameter values from user via menu
c
          include "/home/kanga/dgl3dpg/fortran/include/stack_menu.h"
c
      do while (param.NE.99)
          call display_menu()
          print*
          print 4
          read*,param
          print*
          if (param.EQ.1) then
                print*,'enter new input pathname :'
                read 5,ipath
          elseif (param.EQ.2) then
                print*,'enter new output pathname :'
                read 5,opath
          elseif (param.EQ.3) then
                print*,'enter new filename :'
                read 5,iname
          elseif (param.EQ.4) then
                print*,'enter no. traces per stack (max 50) :'
                read 6,stacksiz
          elseif (param.EQ.5) then
                print*,'enter reduction velocity to stack :'
                read 7,stackvel
```

```
      elseif (param.EQ.6) then
              print*,'is data already reduced ? :'
              read 5,war
      elseif (param.EQ.7) then
              print*,'enter first trace number :'
              read 8,ifirst
      elseif (param.EQ.8) then
              print*,'enter last trace number :'
              read 8,ilast
      elseif (param.EQ.9) then
              print*,'enter channel number (0 to ignore) :'
              read 6,channel
      elseif (param.EQ.10) then
              print*,'is trace separation constant ?'
              read 5,flag
      elseif (param.EQ.11) then
              if ((flag(1:1).EQ.'Y').OR.(flag(1:1).EQ.'y')) then
                print*,'enter trace separation :'
                read 7,trsep
              endif
      elseif (param.EQ.12) then
              if ((flag(1:1).EQ.'Y').OR.(flag(1:1).EQ.'y')) then
                print*,'enter trace number of reduction origin'
                read 8,trorig
              else
                print*,'enter positon of reduction origin (km)'
                read 7,xorig
              endif
      endif
    end do
c
  4   format (2X,'Enter number of parameter to alter (99 to continue) : ',$)
  5   format (A)
  6   format (I)
  7   format (F8.2)
  8   format (I4)
c
      oname = iname(1:lnb(iname))//'_stk'
c
      return
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      subroutine menu_default()
c
      include "/home/kanga/dgl3dpg/fortran/include/stack_menu.h"
c
    ipath = './                '
    opath = './                '
    iname = 'l1101_bpf         '
    oname = '                  '
c
    stacksiz = 5
    stackvel = 6.0
c
    war = 'YES'
c ifirst is trace no. at start of loop, ilast is
c trace no. at end
    ifirst = 1
    ilast  = 9999
c
      channel = 0
c
      flag  = 'NO'
      trsep  = 50.0
```

```
      trorig = 1
      xorig  = 0.0
c
      return
      end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      subroutine display_menu()
c
      include "/home/kanga/dgl3dpg/fortran/include/stack_menu.h"
c
      print*,'\f'
      print 200
      print 201,ipath
      print 202,opath
      print 203,iname
      print 204,stacksiz
      print 205,stackvel
      print 206,war
      print 207,ifirst
      print 208,ilast
      print 209,channel
      print 210,flag
      if ((flag(1:1).EQ.'Y').OR.(flag(1:1).EQ.'y')) then
        print 211,trsep
        print 212,trorig
      else
        print 213
        print 214,xorig
      endif
      print 299
c
  200 format(12X,'**************',/,
     &       12X,'* STACKING *',/,
     &       12X,'**************')
  201 format(2X,'1. Input pathname  :',T35,A25)
  202 format(2X,'2. Output pathname :',T35,A25)
  203 format(2X,'3. Filename        :',T35,A20)
  204 format(2X,'4. Stack size      :',T37,I2)
  205 format(2X,'5. Stack velocity  :',T33,F8.2)
  206 format(2X,'6. Use WAR reduction info  :',T37,A3)
  207 format(2X,'7. First trace number      :',T35,I4)
  208 format(2X,'8. Last trace number       :',T35,I4)
  209 format(2X,'9. Channel no.(0 to ignore):',T38,I1)
  210 format(1X,'10. Constant trace spacing  :',T37,A3)
  211 format(1X,'11. Trace separation        :',T35,F6.3)
  212 format(1X,'12. Origin (trace number)   :',T35,I4)
  213 format(1X,'11. ------------------------')
  214 format(1X,'12. Origin (km)             :',T35,F6.3)
  299 format(1X,'\n',1X,'99.              RUN ')
c
      return
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      subroutine stackini(stacksiz)
c
    include "/home/kanga/dgl3dpg/fortran/include/stack.h"
c
      integer*4 stacksiz
c
      do 10 I=1,stacksiz
       stackpos(I)=I
       id(I)=3
   10 continue
```

```
c
        return
        end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
        subroutine stk(work,srate,nsampl)
c
      include "/home/kanga/dave/Flib/include/segytc.h"
c
        include "/home/kanga/dgl3dpg/fortran/include/stack.h"
        include "/home/kanga/dgl3dpg/fortran/include/stack_menu.h"
c
        integer*4 fold, idout, isamp, shift, nsampl
        real*4    work(16000), sum, slow, srate
c
c copy new trace to stack
        call aacopy(work,1,nsampl,stackdat(1,stackpos(1)),1,nsampl)
c
c extract useful info from trace header common block
          if ((flag(1:1).EQ.'Y').OR.(flag(1:1).EQ.'y')) then
                x(stackpos(1)) = (ufldfn-trorig)*trsep
          else
                x(stackpos(1)) = wrange-(xorig*1000.0)
          endif
          id(stackpos(1))=utrtyp
c
c calculate (1/velocity), correcting for wide angle data
c
        slow  = 1.0/(stackvel*1000.0)
c
          if ((war(1:1).EQ.'y').OR.(war(1:1).EQ.'Y')) then
            if (xrvel.NE.0.) slow = (1.0/xrvel)-(1.0/(stackvel*1000.0))
          endif
c
c check fold
          fold=0
          do 20 J=1,stacksiz
            if (id(stackpos(J)).EQ.1) fold=fold+1
 20       continue
          if (fold.EQ.0) then
            idout=3
            goto 99
          else
            idout=1
          endif
c
c shift & sum samples
          do 90 isamp=1,nsampl
           sum=0.0
           do 80 J=1,stacksiz
             if (id(stackpos(J)).NE.1) goto 80
             shift=int((x(stackpos(1))-x(stackpos(J)))*srate*slow)
             if ((isamp+shift.LE.0).OR.(isamp+shift.GE.nsampl).OR.
     &           (stackdat(isamp+shift,stackpos(J)).EQ.999.999)) then
               work(isamp)=0.0
               goto 90
             endif
             sum=sum+stackdat(isamp+shift,stackpos(J))
 80        continue
           work(isamp)=sum/fold
 90       continue
c
c move traces along one place in stack
         do 95 J=1,stacksiz
          stackpos(J)=stackpos(J)+1
```

```
            if (stackpos(J).gt.(stacksiz)) stackpos(J)=1
  95        continue
c
c amend trace header values
  99          unstk=fold
              utrtyp=idout
c
            return
            end
```

10. STATICS

```
      program statics
c
c accepts trace file sequence and reduces to datum at source+receiver
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      integer*4 ierr, ihand, recsz, nsampl
      integer*4 param
      integer*4 itrace, ifirst, ilast, tshift
      character name*25, ipath*25
            real*4   intrace(16000),outtrace(16000), srate
            real*4   wvel, svel, rvel,sdatum, rdatum, wdepth
            real*4     rcvelev, srcdepth, rshift, wshift
c
      call dlsini()
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c defaults
c
      ipath = './                '
      name = 'l1g128_8           '
c
c ifirst is trace no. at start of loop, ilast is
c trace no. at end
      ifirst = 1
      ilast  = 5000
      itrace = 1
c
            wvel  = 1.4
            svel  = 5.8
            rvel  = 5.8
            trvel = 6.5
            sdatum = 150.0
            rdatum = 0.0
            wdepth = 50.0
            srcdepth=7.5
            rcvelev=10.0
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c get parameter values from user via menu
c
      do while (param.ne.99)
            call menu(ipath,name,ifirst,ilast,wvel,svel,trvel,
     &                sdatum,rdatum,rvel)
            print*
            print 4
            read*,param
            print*
            if (param.EQ.1) then
                  print*,'enter new  pathname :'
                  read 5,ipath
            elseif (param.EQ.2) then
                  print*,'enter new filename :'
                  read 5,name
            elseif (param.EQ.3) then
                  print*,'enter first trace number :'
                  read 8,ifirst
            elseif (param.EQ.4) then
                  print*,'enter last trace number :'
                  read 8,ilast
            elseif (param.EQ.5) then
                  print*,'enter water layer velocity :'
```

```
                read 6,wvel
            elseif (param.EQ.6) then
                print*,'enter sea bed velocity :'
                read 6,svel
            elseif (param.EQ.7) then
                print*,'enter target arrival velocity :'
                read 6,trvel
            elseif (param.EQ.8) then
                print*,'enter source datum depth :'
                read 6,sdatum
            elseif (param.EQ.9) then
                print*,'enter receiver datum elevation :'
                read 6,rdatum
            elseif (param.EQ.10) then
                print*,'enter receiver rock velocity :'
                read 6,rvel
            elseif (param.EQ.90) then
                call report('static correction','menu','I',
     &          'user requested abort')
                call exit(1)
            endif
        enddo
c
    4   format (2X,'Enter number of parameter to alter (99 to continue) : ',$)
    5   format (A)
    6   format (F5.2)
    8   format (I4)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c open  files and read segy header
c
        call optdat (ipath,name,0,recsz,ihand,ierr)
c
        if(ierr.LT.0) then
          call report('statics','dataset','F',
     &  ' Cannot find all dataset files')
          call exit(1)
        endif
c
        call rhead(ihand,0,ierr)
c
        if (ierr.LT.0) then
          call report('statics','dataset','F',
     &  ' Unable to read reel header file')
          call exit(1)
        endif
c
        nsampl = rhinsm
        if (rhisr.NE.0) srate  = 1.0E6/rhisr
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c find first trace
c
        do while (ufldfn.lt.ifirst)
          call rtdh(ihand,itrace,ierr)
          if (ierr.LT.0) then
            call report('statics','dataset','F',
     &  ' Unable to read trace header file')
            call exit(1)
          endif
          if (ierr.EQ.1) then
            call report('statics','dataset','F',
     &  ' shot point range doesn"t exist')
            call exit(1)
          endif
          itrace=itrace+1
```

```
c
          call razclk()
          call flush(6)
c
         end do
         itrace=itrace-2
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c MAIN LOOP
c
     do 10 while (.true.)
          if (ufldfn.GE.ilast) goto 11
          itrace=itrace+1
c         print 1000,itrace
          call razout('Trace : ',itrace)
          call flush(6)
c
c read in trace header to common block
          call rtdh(ihand,itrace,ierr)
          if (ierr.LT.0) then
           call report('statics','dataset','F',
   &   ' Unable to read trace header file')
           call exit(1)
          endif
c
c check for end of file
          if (ierr.GT.0) then
           call report('statics','dataset','I',
   &    ' End of file reached')
           goto 11
          endif
c
c check for correct sample no. and rate
          if (unsamp.NE.0) nsampl = unsamp
          if (wsr.NE.0) srate = 1.0E6/wsr
c
c read in data
          call rtrd (ihand,itrace,intrace,nsampl,ierr)
          if (ierr.LT.0) then
           call report('statics','dataset','F',
   &   ' Unable to read trace data file ')
           call exit(1)
          endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c do static shift
c
c if water depth defined in trace header then set new water depth
c otherwise keep same waterdepth as before
          if (wsrcwd.ne.0.0) then
           wdepth = wsrcwd
          endif
c do same with source depth and receiver elevation
c NB source depth seems to be 10 times too large in PDAS data
          if (wsrcdp.ne.0.0) then
           srcdepth=wsrcdp
          endif
          if (wrcvge.ne.0.0) then
           rcvelev=wrcvge
          endif
c
          wshift=statshift(wdepth-srcdepth,sdatum-srcdepth,wvel,svel,trvel)
          rshift=statshift(0.0,rcvelev-rdatum,wvel,rvel,trvel)
          tshift = int(srate*(wshift+rshift))
          call aacopy(intrace,tshift,nsampl,outtrace,1,
   &     (nsampl-tshift+1))
```

```
c pad out with blanks
            do i=(nsampl-tshift+1),nsampl
             outtrace(i)=0.0
            enddo
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c amend trace header
c
          wrcvde=rdatum
          wsrcde=sdatum
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c write to file
c
            call wtdh(ihand,itrace,ierr)
            if (ierr.LT.0) then
             call report('statics','dataset','F',
     &    ' Unable to write to trace header file')
              call exit(1)
            endif
c
            call wtrd (ihand,itrace,outtrace,nsampl,ierr)
            if (ierr.LT.0) then
             call report('segy','trace data','F',
     &    ' Unable to write to file')
              call cltdat(ihand)
              call exit(1)
            endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
  10  continue
  11  continue
      call cltdat(ihand)
c
 1000  format(1X,'Trace : ',I5)
c
      stop
      end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
          subroutine menu(pathi,name,ifirst,ilast,wvel,svel,trvel,
     &                sdatum,rdatum,rvel)
c
          character*(*) pathi, name
          integer  ifirst, ilast
          real*4   wvel,svel,trvel,sdatum,rdatum,rvel
c
         print*,'\f'
         print 200
         print 201,pathi
         print 202,name
         print 203,ifirst
         print 204,ilast
         print 205,wvel
        print 206,svel
         print 207,trvel
         print 208,sdatum
         print 209,rdatum
         print 210,rvel
         print 290
         print 299
         print 300
c
  200     format(5X,'*******************************',/,
```

```
     &     5X,'*BATHYMETRIC STATIC CORRECTION*',/,
     &     5X,'*******************************')
  201    format(2X,'1. Input pathname :',T45,A25)
  202    format(2X,'2. Filename       :',T45,A20)
  203    format(2X,'3. First trace number          :', T45,I4)
  204    format(2X,'4. Last trace number           :',T45,I4)
  205    format(2X,'5. Water velocity (km/s)       :',T44,F5.2)
  206    format(2X,'6. Sea bed velocity (km/s)     :',T44,F5.2)
  207    format(2X,'7. Target arrival velocity (km/s) :',T44,F5.2)
  208    format(2X,'8. Source datum depth (m)      :',T43,F6.2)
  209    format(2X,'9. Receiver datum elevation (m)  :',T43,F6.2)
  210    format(1X,'10. Receiver rock velocity (km/s) :',T44,F5.2)
  290    format(1X,'\n',1X,'90           ABORT')
  299    format(1X,'\n',1X,'99.          RUN ')
c
  300    format(1X,'\n',1X,'Warning: original data will be overwritten')
c
        return
        end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
        function statshift(depth,datum,vel1,vel2,trvel)
c
c       returns static correction (s) to datum level at or below
c       non-uniform interface
c
c       depth = depth to interface
c       datum = depth to datum level
c       vel1  = constant velocity above interface
c       vel2  = constant velocity below
c       trvel  = target arrival apparent velocity
c
        real*4 depth,datum,vel1,vel2,trvel
        real*4 shift1, shift2
c
        shift1=depth*sqrt(trvel**2-vel1**2)/vel1
        shift2=(datum-depth)*sqrt(trvel**2-vel2**2)/vel2
        statshift=(shift1+shift2)/(trvel*1000.0)
c
        return
        end
```

11. TRACEBIN

```
      program tracebin
c
c     reads in a trace file sequence, bins it and writes it out to
c        a binned dataset, stacking with any trace already present
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      integer*4 ihand, ohand, ierr, irecsz, orecsz
     integer*4 redvel, tstart, id
      integer*4 itrace,otrace, nsampl, isamp
      character*32 iname, oname, ipath, opath, filnam
      real*4  newtrace(16000), srate, bintrace(16000)
      real*4  minoff, binlen, wdist, delay, sintval
c
c       minoff=offset of centre of nearest bin
c       binlen=length of bin
c       wdist =distance to centre of bin(integer multiple of binlen)
c
      call dlsini()
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c default and initial values
c
      ipath ='./                    '
      opath ='./                    '
c
      itrace= 0
c
      write(*,'(A,$)') 'input pathname : '
      read 5,ipath
      write(*,'(A,$)') 'name of dataset to bin : '
      read 5,iname
      write(*,'(A,$)') 'output pathname : '
      read 5,opath
      write(*,'(A,$)') 'name of binned dataset : '
      read 5,oname
c
5     format (A)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c open input files
c
      call optdat (ipath,iname,0,irecsz,ihand,ierr)
      if(ierr.LT.0) then
        print *, iname
        call report('trace_bin','dataset','F',
     &  ' Cannot find all dataset files')
        call exit(1)
      endif
c
c read file header
c
      call rhead(ihand,0,ierr)
      if (ierr.LT.0) then
        call report('trace_bin','dataset','F',
     &  ' Unable to read reel header file')
        call exit(1)
      endif
      nsampl = rhinsm
      if (rhisr.NE.0) srate = 1.0E6/rhisr
c
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c IF binned dataset already exists, open it and read or calculate
c binning parameters
c
      call bldfil(opath,oname,'bfh ',filnam)
      if (exists(filnam)) then
c open old dataset
       call optdat(opath,oname,0,orecsz,ohand,ierr)
       if(ierr.lt.0)then
            print*,oname
        call report('trace_bin','dataset','F',
   &    ' open failed - can"t find all data files')
        call exit(1)
           elseif(irecsz.ne.orecsz) then
            call report('trace_bin','dataset','F',
   &       ' record lengths not compatible')
        call exit(1)
       endif
c read in parameters from trace headers
           call rtdh(ohand,1,ierr)
           if (ierr.LT.0) then
            call report('trace_bin','dataset','F',
   &    ' Unable to read trace header file[1]')
            call exit(1)
           endif
           minoff=wrange
c
           call rtdh(ohand,2,ierr)
           if (ierr.LT.0) then
            call report('trace_bin','dataset','F',
   &    ' Unable to read trace header file[2]')
            call exit(1)
           endif
           binlen=wrange-minoff
c
           if (binlen.eq.0) then
            call report('trace_bin','dataset','F',
   &       ' Trace header ranges not defined ')
            call exit(1)
           endif
c
c ELSE open new dataset
      else
       call optdat(opath,oname,1,irecsz,ohand,ierr)
       if(ierr.lt.0)then
        call report('trace_bin','dataset','F',
   &    ' create&open failed - disc space? or already exists')
        call exit(1)
       endif
c get parameters from user
           print*,'enter minimum offset,binlength (m)'
           read*,minoff,binlen
c
c write out file header
       call whead(ohand,0,ierr)
       if (ierr.LT.0) then
            call report('trace_bin','dataset','F',
   &    ' Unable to write reel header file')
            call exit(1)
       endif
c ENDIF
         endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
```

```
c
c        ***************
c        ** main loop **
c        ***************
c
     do 100 while (.true.)
          itrace=itrace+1
          call razout('binning trace ',itrace)
          call flush(6)
c
c read in trace header to common block
c
          call rtdh(ihand,itrace,ierr)
          if (ierr.LT.0) then
            call report('trace_bin','dataset','F',
     &    ' Unable to read trace header file[3]')
            call exit(1)
          endif
c
c check for end of file
c
          if (ierr.GT.0) then
            call report('trace_bin','dataset','I',
     &    ' End of file reached')
            goto 110
          endif
c
c check for non-seismic traces
          if (utrtyp.ne.1) then
            goto 100
          endif
c
          otrace=int((wrange-minoff)/binlen +1.5)
          if (otrace.le.0) then
            goto 100
          endif
          wdist=minoff+binlen*(otrace-1)
          delay=wdist/xrvel - (xrtoff/1000)
          id=utrtyp
          sintval=wsr
          redvel=xrvel
          tstart=xrtoff
c
c check for correct sample no. and rate **
c
          if (nsampl.EQ.0) nsampl = unsamp
          if (srate.EQ.0.AND.wsr.NE.0) srate = 1.0E6/wsr
c
c read in corresponding bin trace header,clearing common block first
c
          call cltdh()
          call rtdh(ohand,otrace,ierr)
          if (ierr.LT.0) then
            call report('trace_bin','dataset','F',
     &    ' Unable to read trace header file[4]')
            call exit(1)
          endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c read in data
c
          call rtrd (ihand,itrace,newtrace,nsampl,ierr)
          if (ierr.LT.0) then
            call report('trace_bin','dataset','F',
     &    ' Unable to read trace data file ')
```

```
            call exit(1)
           endif
c
           call rtrd (ohand,otrace,bintrace,nsampl,ierr)
           if (ierr.LT.0) then
            call report('trace_bin','dataset','F',
     &     ' Unable to read trace data file ')
            call exit(1)
           endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c stack the two traces
c
           do 90  isamp=1,nsampl
            if ((bintrace(isamp).ne.999.999).and.
     &                (newtrace(isamp).ne.999.999)) then
             bintrace(isamp)=
     &       (newtrace(isamp)+bintrace(isamp)*unstk)/(unstk+1)
            elseif(bintrace(isamp).eq.999.999) then
             bintrace(isamp)=newtrace(isamp)
            endif
  90       continue
c
c amend the header
           ufldfn = otrace
           ufldtn = otrace
           ucdp = 1
           utrace = 1
           utrtyp = id
           unstk = unstk+1
           wrange = wdist
           unsamp = nsampl
           wsr = sintval
           xdelt = int(delay)
           udelt = int((delay-xdelt)*1000)
           xrvel=redvel
           xrtoff=tstart
           xevecd = 'BIN '
c
c write to bin file
c
           call wtrd (ohand,otrace,bintrace,nsampl,ierr)
           if (ierr.LT.0) then
            call report('segy','trace data','F',
     &     ' Unable to write to file')
            call cltdat(ohand)
            call exit(1)
           endif
c
           call wtdh(ohand,otrace,ierr)
           if (ierr.LT.0) then
            call report('segy','trace data','F',
     &     ' Unable to write to file')
            call cltdat(ohand)
            call exit(1)
           endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
  100     continue
c*****************END OF MAIN LOOP**********
  110     continue
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
```

```
c check that all trace headers in bin file have been written to
c
        i=0
        do while(.true.)
         i=i+1
         call rtdh(ohand,i,ierr)
         if (ierr.LT.0) then
          call report('trace_bin','dataset','F',
     &   ' Unable to read trace header file[5]')
          call exit(1)
         endif
c
         if (ierr.GT.0) then
          call report('trace_bin','dataset','I',
     &   ' End of file reached')
          goto 120
         endif
c
         if (ufldfn.eq.0.and.ufldtn.eq.0) then
          ufldfn=i
          ufldtn=i
          wrange=minoff+(i-1)*binlen
c
          call wtdh(ohand,i,ierr)
          if (ierr.LT.0) then
           call report('segy','trace data','F',
     &    ' Unable to write to file')
            call cltdat(ohand)
            call exit(1)
           endif
         endif
c
        enddo

120     continue
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
     call cltdat(ihand)
     call cltdat(ohand)
c
     stop
     end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
```

12. TRANSFORM

```
      program transform
c
c     Rotation, translation and scaling of coordinates from
c     digitising table for use with beam87.
c     N.B. 1st 3 coord pairs in input file must be bottom left,
c     bottom right, and top left corners respectively.
c
      include '/home/kanga/dave/Flib/include/functions.h'
      include '/home/kanga/dave/Flib/include/dls.h'
c
      real t0, t1, d0, d1, t, d
      real a(3), b(3), x, y, vel
      real angle, dscale, tscale, SINA, COSA
      integer idummy, ierr
      character iname*20, oname*20, path*25, ifn*45, ofn*45, join
c
      call dlsini()
      call getparam(path,iname,oname,d0,d1,t0,t1,vel)
c
      join='/'
      if (path(1:1).eq.' ') join=' '
      ifn=path(1:lnb(path))//join(1:lnb(join))//iname
      ofn=path(1:lnb(path))//join(1:lnb(join))//oname
c
      open(unit=10,file=ifn(1:lnb(ifn)),status='OLD')
      open(unit=11,file=ofn(1:lnb(ofn)),status='NEW')
c
      do i=1,3
       read(10,100),a(i),b(i),dummy
      end do
c
      angle=atan(((b(1)-b(2))/(a(2)-a(1))+(a(1)-a(3))/(b(1)-b(3)))/2)
      SINA=sin(angle)
      COSA=cos(angle)
      dscale=(d1-d0)/((a(2)-a(1))*cos(angle)-(b(2)-b(1))*sin(angle))
      tscale=(t1-t0)/((a(3)-a(1))*sin(angle)+(b(3)-b(1))*cos(angle))
c
      do while (.true.)
       do i=1,4
        read(10,100,iostat=ierr),x,y,idummy
        if (ierr.lt.0) then
         print*,'end of file reached'
         goto 50
        endif
        d=dscale*((x-a(1))*COSA-(y-b(1))*SINA)+d0
        t=tscale*((x-a(1))*SINA+(y-b(1))*COSA)+t0
c unreduce data if necessary
        if (vel.ne.0) t=t+d/vel
        write(11,110) d,t
       end do
       write(11,*) ' '
      end do
c
 50   close(unit=11)
      close(unit=10)
c
 100  format(f8.3,1x,f8.3,1x,i2)
 110  format(f10.5,f10.5,$)
c
      end
```

```
c
      subroutine getparam(path,iname,oname,d0,d1,t0,t1,vel)
c
      real t0, d0, t1, d1, vel
      character iname*20, oname*20, path*25, red*3
c
      print*,'\f** Transformation of coordinates **\n'
      print 10
      read(*,'(A25)'),path
      print 20
      read(*,'(A20)'),iname
      print 30
      read(*,'(A20)'),oname
      print 40
      read(*,'(F8.0)'),d0
      print 50
      read(*,'(F8.0)'),d1
      print 60
      read(*,'(F5.0)'),t0
      print 70
      read(*,'(F5.0)'),t1
      print 80
      read(*,'(A3)'),red
      if ((red(1:1).eq.'y').or.(red(1:1).eq.'Y')) then
       print 90
       read(*,'(F5.2)'),vel
      else
       vel=0.0
      endif
c
   10 format(' path name : ',$)
   20 format(' input filename : ',$)
   30 format(' output filename : ',$)
   40 format(' left side offset : ',$)
   50 format(' right side offset : ',$)
   60 format(' bottom time : ',$)
   70 format(' top time : ',$)
   80 format(' is data already reduced ? ',$)
   90 format(' reduction velocity used : ',$)
c
      return
      end
```

13. WATERD

```
      program  waterd

c reads header file and writes out the water depth to screen,
c distance etc for every nrecth trace.
c
      include "/home/kanga/dave/Flib/include/rhicom.h"
      include "/home/kanga/dave/Flib/include/segytc.h"
      include "/home/kanga/dave/Flib/include/functions.h"
      include "/home/kanga/dave/Flib/include/dls.h"
c
      integer*4 ierr, ihand, recsz
      integer*4 itrace, ifirst, ilast, nrec,otrace
      integer*4 nsampl
      real*4    srate
      character iname*20, oname*24, ipath*25, opath*25
c
      call dlsini()
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c defaults
      ipath = '.                    '
      opath = '.                    '
c ifirst is trace no. at start of loop, ilast is
c trace no. at end, and nrec is step size
      otrace = 1
      ifirst = 1
      ilast  = 4000
      nrec   = 100
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c get parameter values from user via menu
c
            print*,'enter trace step :'
                  read 5,nrec
  5         format (i5)
c
            print *, 'enter filename'
            read 6, iname
c           iname = 'l6bs5_bpf'
  6     format (a)
c
            oname = iname(1:lnb(iname))//'_hdat'
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c open input and output files
c
            call optdat (ipath,iname,0,recsz,ihand,ierr)
            if(ierr.LT.0) then
               call report('segy','dataset','F',
     &  ' Cannot find all dataset files')
               call exit(1)
            endif
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c read segy header
c
            call rhead(ihand,0,ierr)
            if (ierr.LT.0) then
               call report('segy','dataset','F',
     &   ' Unable to read reel header file')
               call exit(1)
            endif
c
            nsampl = rhinsm
            if (rhisr.NE.0) srate = 1.0E6/rhisr
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
c
      do 10 itrace=ifirst,ilast,nrec
c
c read in trace header to common block
            call rtdh(ihand,itrace,ierr)
            if (ierr.LT.0) then
              call report('segy','dataset','F',
     &    ' Unable to read trace header file')
              call exit(1)
            endif
c
c check for end of file
            if (ierr.GT.0) then
              call report('segy','dataset','I',
     &    ' End of file reached')
              goto 11
            endif
c
            if (nsampl.EQ.0) nsampl = unsamp
            if (srate.EQ.0.AND.wsr.NE.0) srate = 1.0E6/wsr
c
           print 1001, ufldfn,wrange/1000,wsrcwd
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
  10  continue
  11  continue
c
 1001  format(i6,2x,2(f10.3,1x))
c
      stop
      end
```

Appendix B : Routines used in above programs

Contents:

```
ADDSUFF
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c create filename by adding suffix to user-defined
c name (padded with ' 's) and giving length of new
c filename (without ' 's)
c
      subroutine addsuff(name,suffix,newname,length)
c
      character*(*) name,suffix,newname
      integer m,length
c
      newname=name
      m=index(newname,' ')
      newname(m:)=suffix//'
      length=index(newname,' ') - 1
      if (length.EQ.-1) length=len(newname)
c
      return
      end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc


NORM
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
    subroutine norm(x,lx,lim,scale)
c
c   normallization to +,- lim for real*4 data,
c   scaling factor returned
c
    real*4 x(lx),lim,scale
c
c determine max
c
    call minmax(lx,x,amin,amax)
    if((amax-amin).eq.0.0) then
      scale=0.0
    else
      scale=lim/max(abs(amax),abs(amin))
    endif
    call avmul(lx,x,scale)
    return
    end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc


TAPER
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine taper(lx1,data,l1)
c
c perform cosine taper on both ends of real array over l1 samples
c
c       data(lx1) = data array
c       l1        = taper length
c
      integer*4 lx1,l1
      real*4   data(lx1)
      real*4   pi,d1,f1,arg
      parameter(pi=3.1415927)
c
      d1=pi/real(l1)
c
       do 20 i=1,lx1
        if (i.le.l1) then
          arg=d1*real(i-1)
```

```
            elseif (i.ge.(lx1-l1)) then
             arg=d1*real(lx1-i+1)
            else
             arg=pi
            endif
            f1=0.5*(1-(cos(arg)))
c
            data(i)=data(i)*f1
c
20          continue
c
          return
          end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```


TAPER_2

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
          subroutine taper_2(lx1,data,l1,l2)
c
c perform cosine taper on both ends of real array
c         data(lx1) = data array
c         l1        = taper length at start of array
c         l2        = taper length at end of array
c
          integer*4 lx1,l1,l2
          real*4    data(lx1)
          real*4    pi,d1,d2,f,arg
          parameter(pi=3.1415927)
c
          d1=pi/real(l1)
          d2=pi/real(l2)
c
          do 20 i=1,lx1
           if (i.le.l1) then
            arg=d1*real(i-1)
           elseif (i.ge.(lx1-l2)) then
            arg=d2*real(lx1-i+1)
           else
            arg=pi
           endif
           f=0.5*(1-(cos(arg)))
           data(i)=data(i)*f
20         continue
c
          return
          end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

TAPER2D

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
          subroutine taper2d(lx1,lx2,data,l1,l2)
c
c perform cosine taper on 2d real array
c
          integer*4 lx1,lx2,l1,l2
          real*4    data(lx1,lx2)
c
          real*4    pi,d1,d2,f1,f2,arg
          parameter(pi=3.1415927)
c
          d1=pi/real(l1)
          d2=pi/real(l2)
```

```
REVERSE2D
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine reverse2d(data,l1,l2)
c
c reverse real*4  array in 2nd dimension
c       <----l2----->
c  ^    2 2 0 1 1              1 1 0 2 2
c  |    2 2 0 1 1              1 1 0 2 2
c  l1   2 2 0 1 1  ========>   1 1 0 2 2
c  |    2 2 0 1 1              1 1 0 2 2
c  v    2 2 0 1 1              1 1 0 2 2
c
      integer*4 l1,l2
      real*4    data(l1,l2), dummy
c
      do j=1,l2/2
       do i=1,l1
        dummy=data(i,j)
        data(i,j)=data(i,l2-j+1)
        data(i,l2-j+1)=dummy
       enddo
      enddo
c
      return
      end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
c
      do 10 j=1,lx2
       if (j.le.l2) then
        arg=d2*real(j)
       elseif (j.ge.(lx2-l2)) then
        arg=d2*real(lx2-j)
       else
        arg=pi
       endif
       f2=0.5*(1-(cos(arg)))
       do 20 i=1,lx1
        if (i.le.l1) then
         arg=d1*real(i)
        elseif (i.ge.(lx1-l1)) then
         arg=d1*real(lx1-i)
        else
         arg=pi
        endif
        f1=0.5*(1-(cos(arg)))
c
        data(i,j)=data(i,j)*f1*f2
c
20     continue
10    continue
c
      return
      end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

TRANSPOSE

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine transpose(data,l)
c
c transpose real*4 square array along principle diagonal
c
c       0 1 1 1 1                0 2 2 2 2
c       2 0 1 1 1                1 0 2 2 2
c       2 2 0 1 1  ========>     1 1 0 2 2
c       2 2 2 0 1                1 1 1 0 2
c       2 2 2 2 0                1 1 1 1 0
c
      integer*4 l
      real*4   data(l,l), dummy
c
      do j=1,l
       do i=j,l
        dummy=data(i,j)
        data(i,j)=data(j,i)
        data(j,i)=dummy
       enddo
      enddo
c
      return
      end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

Appendix C: Routines from other sources

D.L.Stephenson:

AACOPY	Real array copy routine
ASKOK	Ask for YES/NO confirmation from user
AZERO	Zero a real array
BPASS	Band pass filter routine (variable window,freq domain)
CLTDAT	Close ISEGY dataset files
DLSINI	Initialise DLS subroutine library
EXISTS	Test whether a file exists
FFTA	Fast Fourier Transform (Cooley-Tukey)
FFT2D	2D Fast Fourier Transform
GBEGIN	Open UNIRAS file + segment
GEND	Close UNIRAS file
KORD2D	Reorder 2D f-k array for increasing k (or vice versa)
LININT	Linear interpolation between points in an array
LNB	Determine last non blank character in a string
MEAN	Calculate mean and remove from a real array
OPTDAT	Open ISEGY dataset files
PAD	Fill a real array with a constant
REPORT	Error report subroutine
RESAMP	Time domain resampling subroutine using sin(x)/x
RHEAD	Read ISEGY reel header file
RTDH	Read ISEGY trace data header from file
RTRD	Read ISEGY trace data from file
USRMES	Print a string on the terminal
WHEAD	Write ISEGY reel header file
WTDH	Write ISEGY trace data header to file
WTRD	Write ISEGY trace data to file

C. Prescott:

RAZCLK	Write rotating bar to screen without scrolling
RAZOUT	Write string and integer to screen without scrolling

Appendix D: Programs from other sources

C. Peirce / C.Prescott:

DAZZLE	Seismic data plotting program (UNIRAS graphics)

P. A. Matthews:

CONVERT	Convert PDAS-type seismic data files (IBM format) to ISEGY format (SUN format)

V. Cerveny:

BEAM87	Forward modelling package using Gaussian Beam approximation for wide angle seismic data (adapted for UNIRAS graphics by C. Prescott and P. A. Matthews; adapted for normal incidence by P. A. Matthews)

D. L. Stephenson:

SEGY	Ready SEGY reel data and store in ISEGY format.

M.H.P.Bott

GRAV	Two dimensional gravity modelling program

GLOBAL LAB

GLOBAL LAB	Digitisation and data display package (Macros written by R. E. Long)

Appendix E: Large scale data record sections (1:400000)

1: Station 101P - Band pass filtered
- Stacked @ 6.5 km s^{-1} (5 trace)

2: Station F01A - Band pass filtered

3: Station 1A - Band pass filtered

4: Station 5A - Band pass filtered
- Binned (bin width 75 m)