Lappeenrannan teknillinen korkeakoulu

**Lappeenranta University of Technology**

**Tuomo Kauranne**

# INTRODUCING PARALLEL COMPUTERS INTO OPERATIONAL WEATHER FORECASTING

*Thesis for the degree of Doctor of Philosophy to be presented with due permission for public examination and criticism in the Auditorium 1383 of the Lappeenranta University of Technology, Lappeenranta, Finland, on the 20th of December 2002, at noon.*

**Acta Universitatis**
**Lappeenrantaensis**
**142**

Supervisor Professor Heikki Haario
Lappeenranta University of Technology
Department of Applied Mathematics
Lappeenranta, Finland

Opponents Dr. Heikki Järvinen
Finnish Meteorological Institute
Helsinki, Finland

Dr. Jussi Heikonen
Center for Scientific Computing
Espoo, Finland

**ISBN 951-764-715-8**

**ISSN 1456-4491**

**Lappeenrannan teknillinen korkeakoulu**

**Digipaino 2002**

# Abstract

Tuomo Kauranne

INTRODUCING MASSIVELY PARALLEL COMPUTERS INTO OPERATIONAL WEATHER FORECASTING

Lappeenranta, 2002
65 p.

Acta Universitatis Lappeenrantaensis 142
Diss. Lappeenranta University of Technology

ISBN 951-764-715-8
ISSN 1456-4491

Numerical weather prediction and climate simulation have been among the computationally most demanding applications of high performance computing ever since they were started in the 1950's. Since the 1980's, the most powerful computers have featured an ever larger number of processors. By the early 2000's, this number is often several thousand. An operational weather model must use all these processors in a highly coordinated fashion. The critical resource in running such models is not computation, but the amount of necessary communication between the processors. The communication capacity of parallel computers often falls far short of their computational power.

The articles in this thesis cover fourteen years of research into how to harness thousands of processors on a single weather forecast or climate simulation, so that the application can benefit as much as possible from the power of parallel high performance computers. The results attained in these articles have already been widely applied, so that currently most of the organizations that carry out global weather forecasting or climate simulation anywhere in the world use methods introduced in them. Some further studies extend parallelization opportunities into other parts of the weather forecasting environment, in particular to data assimilation of satellite observations.

Keywords: Parallel computing, Numerical weather prediction, Climate simulation, Data Assimilation.

UDC 004.272 : 551.509.3

# Contents

# Summary

The current introduction and the articles that comprise the thesis span more than ten years of research into the use of parallel supercomputers in operational weather forecasting between 1988 and 2002. The articles represent a sequence of studies from theoretical estimates, through parallelization tests with simplified atmospheric models, to a wholesale port of an operational weather model onto several massively parallel computers.

Parallel to the studies of how to parallelize operational forecast models, a theoretical analysis has been carried out on how to parallelize the most effective modern data assimilation method, variational assimilation. This analysis is supported by some numerical tests with simple one dimensional atmospheric models.

This particular order of the studies was mandated by the operational status and plans of the operational weather forecasting centres, for which the studies were carried out, in particular the European Centre for Medium-range Weather Forecasts (ECMWF) in Reading, UK, and the Finnish Meteorological Institute (FMI) and the Center for Scientific Computing (CSC) in Helsinki and Espoo in Finland, respectively. The more theoretical studies have been conducted, partly afterwards, at the University of Joensuu, at the Italian University Supercomputing Centre CINECA in Bologna, and at the Lappeenranta University of Technology.

During these fourteen years, the results of the earlier studies in the thesis have gained widespread acceptance in the operational weather forecasting community. In particular, the principal innovation of the first articles, the *transposition strategy* to implement data communications on a distributed memory supercomputer, has won unanimous acceptance worldwide among the operational weather forecasting centres that carry out global forecasting with spectral models. It has become almost as popular also among the centres running Limited Area Models, as it handles very well the semi-implicit time stepping schemes employed in these grid point models.

Weather and climate forecasting centers that use the transposition strategy include ECMWF, FMI, Meteo France, Max Planck Institute in Hamburg, the U. S. national weather forecasting centre NPEC in Washington, Fleet Navy Oceanography Center in Monterrey, Japanese Meteorological Agency and the Swedish and Norvegian meteorological institutes. The German weather service DWD also did, for as long as they carried out global weather forecasts. The most recent adopter of the transposition strategy is the Earth Simulator in Japan, which is the most powerful supercomputer

in the world by far in 2002, and is being used exlusively for simulating the climate at very high resolution.

I believe it is not too bold to claim that the studies reported in the current thesis were decisive for the adoption of massively parallel, distributed memory supercomputers in operational weather forecasting, see e.g. the articles in the Proceedings of the ECMWF workshops on the Use of Parallel Computing in Meteorology since 1994: *Hoffmann and Kreitz (1995), Hoffmann and Kreitz (1997), Zwiefelhofer and Kreitz (1999)* and *Zwiefelhofer and Kreitz (2001).*

The studies have contributed to the decision of both ECMWF, and FMI and CSC, to purchase a distributed memory supercomputer for operational use in the mid-1990's. They were the first centres in the world to do so. Both ECMWF and FMI have successfully run their operational forecasting activities on massively parallel, distributed memory computers ever since.

Many other centres have followed suite since 1996, so that as of this writing, in late 2002, an operational weather forecasting centre still using a shared memory vector supercomputer is a rare exception. Most centres are into their second or third generation massively parallel computers. Most of them employ workstation cluster based Non-Uniform Memory Access (NUMA) type computers, that in some cases have vector processors instead, from manufacturers such as IBM, NEC, Cray, SGI or Fujitsu.

As to the theoretical studies on variational assimilation in the current thesis, both for operational weather forecasts and for aeronomic assimilation, their true impact still resides in the future. These studies were originally aimed at alleviating a threat to parallel scalability, imposed by an inherent seriality in the variational data assimilation cycle that has been adopted by most weather forecasting centres. This threat has been eased by a shift from a single high resolution forecast to an ensemble of moderate resolution forecasts, as well as by the fact that the power of parallel supercomputers has increased faster than the computational complexity of a single operational forecast at current operational resolution.

However, the theoretical properties of variational assimilation are important and interesting in their own right. Variational assimilation automatically produces a degree of error control to the forecast in the form of an estimate to the error covariance matrix. Much current algorithmic development in the meteorological community is directed towards how best to utilize such information in the form of simplified Kalman filtering procedures, see e.g. *Fisher (1998).*

The research in the article **(IV)** in the thesis from 1992, and in its very last article **(VI)** from 2002, attempts to develop and partly verify highly efficient ways to produce such error estimates in the course of standard variational data assimilation. A second goal is to provide a strategy to deploy such algorithms efficiently on massively parallel computers in order to eventually also overcome the inherent serial bottleneck in variational assimilation.

The author hopes and believes that these results, too, will eventually show their worth in an operational context, following the success of the earlier studies.

# Acknowledgements

As will have become evident from the narrative above, the research reported in this thesis has been very much a team effort, and would have been totally unthinkable otherwise. I wish therefore to express my deepest gratitude to the key people in my two research teams that have made all these results possible: Saulo Barros and Lars Isaksen on the ECMWF Genesis team, and Juha Oinonen, Sami Saarinen and Jarkko Hietaniemi on the Finnish parallel HIRLAM team. It is both a personal and a professional pleasure to have had the privilege to lead such exceptional teams.

I also wish to thank my supervisor, Heikki Haario from the Lappeenranta University of Technology, and my opponents Heikki Järvinen from the Finnish Meteorological Institute and formerly of ECMWF, and Jussi Heikonen from the Center for Scientific Computing, for a very constructive attitude towards my work. They have all provided discreet guidance to me on how to improve the thesis, while always showing respect towards the efforts needed. I hope that the final outcome shows, that I have also tried to follow their sound advice.

I have been very fortunate with the organisations and peers I have had the opportunity to work for. I wish to thank Matti Ihamuotila and Juhani Käpyaho of the Center for Scientific Computing, Geerd-R. Hoffmann and Adrian Simmons of ECMWF, Pekka Neittaanmäki of the University of Jyväskylä, and Matti Heiliö and Jorma Mattila of the Lappeenranta University of Technology for allowing me to throw in my two nickles' worth for the organisations, divisions, departments and teams they lead. Not only have they allowed me to do my job very independently, but they have also shown, by their own example, how one should lead a succesful team of experts: not by coercion, but by spirit.

Ever since returning from the United Kingdom, I have enjoyed the lively working environments at the University of Joensuu and at the Lappeenranta University of Technology. It has also been a great pleasure to put part of this thesis together in the relaxed and intellectually inspiring environment of the Italian academic supercomputer centre CINECA in Bologna. I am very grateful for having been allowed to participate in their ICARUS programme, that has granted me the possibility to invest a crucial few concentrated weeks of work into the finishing touch to the thesis. I doubt if the work would ever have reached its final stage without such an option.

I wish to thank, for reasons too numerous to mention, the following people that have had a crucial impact on one turn or another of the work conducted, often by providing a key component of labour, code, bright ideas, funding, friendship or otherwise, without which progress would have been difficult: Eric Sevault, Daniel Söderman, John Anderson, Olli Serimaa, Juha Hulkkonen, Richard Reuter, Peter Lönnberg, Nils Gustafsson, Roar Skålin, David Dent, Clive Temperton, John Adams, Paul Swarztrauber, Hal Ritchie, Ute Gärtel, Anton Schüller, Simon Tett, Juhani Saastamoinen, Harri Auvinen, Andy Marsh, Sanzio Bassini, Ivan Grossi, Francois-Xavier LeDimet, Mike Navon and Yannick Tremolet.

Many others have generously shared with me their insightful thoughts on matters parallel and meteorological, and I apologize for not being able to mention them all in these acknowledgements. Some more can be found in the acknowledgements attached to each of the articles but mention or no mention, my gratitude still remains the same.

Finally, I wish to express my most personal, profound indebtedness to the people that have shaped my whole life: my parents Aino and Jouko Kauranne, my parents-in-law Bertta and Veikko Turunen and, most of all, my caring wife Aino and my sons Juho and Toivo. It may be that all the results between these covers would have been obtained also without their input, but certainly not by me.

It is to Aino, Juho and Toivo that I wish to dedicate this Thesis.

Joensuu, 21 November 2002

*Tuomo Kauranne*

# 1. Articles comprising the thesis

The thesis consists of the current summary and the following six articles:

**I** **Kauranne, T.**: *Summary of GENESIS work at the European Centre for Medium-range Weather Forecasts (ECMWF).* Parallel computing 20 (1994) 1685 - 1688.

**II** **Kauranne, T. and Barros, S.**: *Scalability estimates of parallel spectral atmospheric models.* Parallel supercomputing in atmospheric science (Hoffmann, G.-R. and Kauranne, T., eds.) 312 - 328. Proceedings of the Fifth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1992. World Scientific 1993.

**III** **Barros, S. and Kauranne, T.**: *On the parallelization of global spectral weather models.* Parallel computing 20 (1994) 1335 - 1356.

**IV** **Kauranne, T.**: *4D Variational assimilation, ensemble forecasting and parallel computing.* Parallel supercomputing in atmospheric science (Hoffmann, G.-R. and Kauranne, T., eds.) 286 - 311. Proceedings of the Fifth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1992. World Scientific 1993.

**V** **Kauranne, T., Oinonen, J., Saarinen, S., Serimaa, O. and Hietaniemi J.**: *The Operational HIRLAM 2 model on parallel computers.* Coming of age. (Hoffmann, G.-R. and Kreitz, N., eds.) 63 - 74. Proceedings of the Sixth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1994. World Scientific 1995.

**VI** **Kauranne, T., Haario, H., Auvinen, H.**: *Parallel variational assimilation in aeronomy.* Research Report no. 81, Department of Information Technology, Lappeenranta University of Technology. Lappeenranta University of Technology 2002.

For the sake of completeness, the entire research effort, of which the current thesis forms just a part, is described briefly in the sections below. Whenever results that fall within the current thesis are discussed, reference is initially made to one of the articles above in boldface Roman numerals. In the subsequent narrative in the same section, the text discusses research within the current thesis, unless otherwise indicated by referencing.

## 1.1 Statement on the author's contribution to joint papers

Regarding the design and carrying out of all the research discussed above, it is appropriate to state in the very beginning that this author's principal role in most of the efforts described was to lead a team of researchers - and other teams collaborating with us - to validating conclusions and to producing efficiently running codes. These activities have helped the organizations employing us in making the right supercomputer purchase decisions. This basic disposition has implied that the author's principal role has been to produce and evaluate ideas and hypotheses, and to supervise the other team members in testing and verifying them by experiments.

The author's principal contribution has consequently been in the 'vague end' of the research: parallelization ideas, preliminary assessments and mathematical estimates, as well as strategic decisions. Most of the 'hard end' work: actual coding, and production of numbers to base conclusions on, has been done by the other team members. Even if it is hard, in such a case as that of the author's, to point to concrete contributions, the impact of abstract work and correct judgement has shown in the speed of reaching correct conclusions, and in efficient code conversion under a tight schedule.

Of the two joint papers with Saulo Barros, (**II** and **III**), the author is principally responsible for the work reported in (**II**), whereas Saulo Barros has had the main responsibility for the work in (**III**). In (**II**), the author - exceptionally - also did all the coding necessary to come up with the estimates. In (**III**), Saulo Barros did the numerical analysis and all the coding for the semi-implicit, and particularly the semi-Lagrangian, time stepping method for shallow water equations, as well as carried out all the benchmark runs. The initial spectral discretization for a Helmholtz solver forming the computational kernel of the semi-implicit time stepping scheme was originally coded by the author, based on transform routines by John Adams and Paul Swarztrauber of NCAR. This code was subsequently rewritten by Saulo Barros on the basis of ECMWF transform routines.

Of the parallelization algorithms, Saulo Barros has invented the load balancing allocation of spectral components to processors, illustrated in Figure 1 of (**III**). The author was the first one to propose and code, in the Helmholtz case, the transposition strategy itself. However, the coding in the shallow water case has been done by Saulo Barros. In (**III**), the author is mainly responsible for the qualitative parallelization analysis in section 6, and of the discussion of the 3D transposition strategy in section 7, as illustrated in Figure 2 of (**III**).

As regards joint work between the author and Juha Oinonen, Sami Saarinen, Jarkko Hietaniemi and Olli Serimaa reported in (**V**), the role of the author was even more prominently that of a team leader. The author did no coding whatsoever, but came up with all the basic principles that the subsequent work was based on: that of adopting the transposition strategy, and how to implement it in the case of a grid point model; that of hiding message passing with MPI in a separate subroutine library; that of insisting on reproducible code (i.e. code that produces bitwise identical results on any number of processors), and how to achieve this; and that of choosing to base input and output on GRIB coded subdomains, so as to have at least some standard to base parallel input and output on, and thereby achieving complete portability of the parallel code. Yet another important target, that was attained, was to have the single processor code identical with the one-processor parallel code, thereby avoiding duplicated code maintenance effort.

The work in the project was so assigned that Sami Saarinen was responsible for the design and implementation the Application Dependent Message Passing Library, behind which all non-standard Fortran parallel code was hidden. Sami Saarinen also invented the pairwise message passing scheduling algorithm, illustrated in Figure 2 of (**V**), that avoids deadlocks during each transposition stage. Juha Oinonen, working under close supervision by the author, rewrote the HIRLAM model code itself, attributing all arrays to be split with the necessary additional indices and dimensions. Jarkko Hietaniemi was responsible for all GRIB related work. Olli Serimaa shared in project design and management, and was liasing the project with the parent organisations.

The paper (**V**) is mostly written by the author, except the illustrations and the tables, and it forms the introductory chapter in the extensive User Guide and Technical Documentation to the parallel HIRLAM, produced jointly by the project team (*Saarinen et al. (1995)*), with Sami Saarinen making the largest contributions in text and polishing the final layout.

The joint article (**VI**) with Heikki Haario and Harri Auvinen has been to a very large extent produced by the author, including much of the programming. The model and the idea on how to test it has been developed by Heikki Haario, with Harri Auvinen responsible for coding it in the Matlab environment, for which I am very grateful. The second model is based on ideas in *Talagrand (1991).* The research on parallel assimilation, on search directions, and the coding required for both, are by the author alone.

# 2. Introduction

The six articles comprising the thesis represent five rather independent studies, each discussing a different problem and using a unique method to solve it. Yet all the studies share a single scientific goal that is motivated by a very practical technical and financial concern. This is the emergence of massively parallel, distributed memory supercomputers onto the scientific computing marketplace since the 1980's.

Parallel computing has received abundant academic attention since 1960's, and the field has evolved into a subdiscipline of computer science in its own right. For the practicing scientific computing organization, however, the issue of parallel computing has been decisively solved only in the late 1990's. By the early 2000's, distributed memory parallel high performance computers are used by virtually all high performance computing centres.

Weather forecasting is a prime example of a high technology field where success is critically dependent on three independently evolving, yet interfering, technologies:

- Algorithm development for solving the underlying partial differential equations and their associated optimal control problems
- Fastest commercially available supercomputers
- Development and maintenance of huge operational program codes - each representing an investment of hundreds of person years - that implement the algorithms on the supercomputers chosen, yet outlive both any individual algorithm and any particular supercomputer

A successful weather forecasting organization must keep abreast of both of the two first mentioned technologies by building upon the third technology, within the constraints imposed by the human and financial resources available. Success scores between weather forecasts produced by different forecasting agencies are compared routinely on a daily basis, and they form the basis for future funding decisions between competing forecasting agencies. Successful adoption of newest computing technology is therefore a mission critical requirement for such organizations.

# 3. Developments in supercomputing

Vector supercomputers turned parallel in the early 1980s and were rapidly deployed in operational weather forecasting and other scientific computing practice. In the ensuing twenty years such shared memory parallel computing has migrated down to client-server architectures and it has become a standard option for boosting the performance of server architectures.

From the application developers' point-of-view this development has been unproblematic, since automatic parallelizing compilers can often be applied on existing sequential codes with a good gain in performance. However, experience so far indicates that shared memory parallelism only works up to a few dozen processors, up to the point when the uniform memory access from all the processors to the shared memory saturates the memory bandwidth, and adding more processors will not increase sustained application performance any more.

In the high end of scientific computing - the field often dubbed "supercomputing" - it has become increasingly apparent that the ability to engage hundreds or even thousands of processors in solving the same problem facilitates tremendous gains in sustained application performance. This gain comes at a premium, though: application codes must be rewritten in order to accommodate the distributed memory architecture that alone facilitates efficient use of such massively parallel computers.

Massively parallel computers are far more sensitive to implementation detail in producing good application performance than shared memory parallel computers. It is not uncommon to see hundredfold performance degradation due to a single mistake in implementing a code on a massively parallel computer.

The best compromise to resolve the dilemma above produced by the markets by the early 2000's are massively parallel computers that consist of loosely coupled distributed memory clusters, each consisting of tightly coupled scalar processors with virtual shared memory. Such an architecture is called NUMA, for Non-Uniform Memory Access. Physically, each cluster comprises typically up to 32 processors with a bus or memory switch that allows fast access to the memories of the other processors, too. Internally, each processor and cluster has one to three cache levels with an even faster access time. The clusters are connected to one another by an optical or electronic switch.

Scalar processors have become very small and consequently very fast in the course of the 1990's. The force behind this trend is the massive development effort invested in designing mass market chips. The clock cycles of mundane PC's have become shorter than those of the fastest vector processors of the previous generation.

With careful shared memory programming that maximizes the reuse of data in cache, the single processor performance of such scalar processors now attains one half of that of vector processors of the early 1990s, and roughly one eighth of that of current vector processors, in sustained performance, but at a fraction of the cost. The vision of the designers of the Transputer in the 1980's has therefore come true: ten thousand chicken are able to beat a horse on a general supercomputing workload.

For large scale scientific codes that need to employ almost all of these ten thousand chicken at once, the most sensitive part of a NUMA computer is the switch of the high performance network that is used to couple the clusters. The difference in capacity between the gross numerical output of a big NUMA computer, and the capacity of the switch to transmit these results across the whole computer, has grown, too.

The difference of memory access on a NUMA computer shows up also on the programming model side. Within the clusters, shared memory programming is viable and a programming framework called OpenMP allows programmers to guide data allocation to the virtual shared memory for improved performance. Across the clusters, explicit distributed memory programming is necessary. This is carried out often with message passing, using the MPI message passing interface standard.

Although the troughput of weather models continues to increase, we may be approaching another ceiling in sustained performance, similar to the one hit by vector supercomputers earlier on, unless further algorithmic innovations go around it, rather than try and break it. The investigations in the current thesis try and point out some directions, in which this could be achieved.

# 4. Developments in numerical weather prediction

For the past forty years, numerical weather prediction has been regarded to be identical with the computer model that solves the underlying partial differential equations. In the course of the years, the differential equations used to model atmospheric dynamics for operational weather forecasting purposes have evolved from a single two-dimensional scalar equation to a range of three-dimensional vector valued systems.

The numerical methods used for solving the systems have also diversified, with grid-point methods, which are based on finite differences, finite elements or finite volumes, remaining dominant in limited area and mesoscale atmospheric models, while spectral discretization methods have become prevalent in global weather and climate models. Time stepping schemes have become increasingly implicit, allowing for computational savings through longer time steps. This evolution mandates the solution of increasingly complex, linear or nonlinear elliptic partial differential equations or systems every time step, and a variety of algorithms have been developed for this.

However, not only has the forecast model become more complex: it is also being used in an ever increasing variety of ways to assist in weather forecasting. Before a numerical weather forecast can proceed, an initial state of the atmosphere must be determined on the basis of weather observations. This process, known as data assimilation, has traditionally been solved with adapted linear statistical techniques independent of the forecast model.

Recent developments reformulate the data assimilation problem as an initial control problem for the forecast model, with the goal of minimizing the deviation of the forecast from the observations over the observation period. The most advanced method that has been widely adopted in operational forecasting practice is called four-dimensional variational data assimilation, often dubbed 4DVAR. It was introduced to operational weather forecasting by *Le Dimet and Talagrand (1986).* 4DVAR is an approximation to the known optimal solution to the above control problem: the Extended Kalman Filter. Some variant of the Kalman Filter is often seen as the logical next step.

It has been often observed that the principal practical problem for published weather forecasts is not any more the average accuracy of the forecast per se, which has improved steadily, hand-in-hand with an increase in numerical resolution facilitated by increasingly fast supercomputers. A more prominent problem is often the dramatic variance in forecast skill subject to different local weather conditions. Atmospheric dynamics are a chaotic dynamical system that displays varying degrees of *predictability* under different initial conditions.

The problem of estimating predictability on a daily basis has been approached via a so-called ensemble forecasting method, in which a family of low resolution forecasts is issued from a perturbed set of initial conditions. The directions and magnitudes of the set of perturbations are determined on the basis of a singular vector analysis of a simplified system of equations, and the ensuing spread of the family of forecasts is used to estimate the sensitivity of the current main forecast.

Such diverse uses of the forecast model, as well as the varying computational characteristics of the model itself, complicate greatly the cost-benefit analysis of the adoption of massively parallel computers. For an insightful overview of the delicate interplay between forecasting needs and the selection between different supercomputer architectures, please see *Simmons (1994)*.

# 5. An overview of the current research effort

The current research set out to resolve the issue of the adoption of massively parallel computers in the case of operational weather forecasting. The research was conducted at the European Centre for Medium-range Weather Forecasts (ECMWF) in Reading between 1988 and 1992, targeting the centre's newest model suite called IFS (Integrated Forecasting System), and at the Center for Scientific Computing, Helsinki University of Technology and the University of Joensuu in 1994, targeting the HIRLAM (HIgh Resolution Limited Area Model) model that has been jointly developed and used by the national weather services in Finland, Sweden, Denmark, Norway, Iceland, the Netherlands, Ireland and Spain. The last part of the effort, targeting parallel off-line assimilation of aeronomic models that simulate chemical processes in the stratosphere, has been conducted at the Lappeenranta University of Technology in 2002.

The work has been carried out in six stages, each of which is briefly described in sections 5.1 to 5.6 below. The goal of this sequence of stages has been the production of a near optimal roadmap for adopting massively parallel, distributed memory computers in operational weather forecasting. A migration of large operational scientific codes to a completely different machine architecture is wrought with risks. Such risks include

- *Poor scalability of parallel code*: Parallel scalability means the relative wall-clock speed gain that an application wins, when an increasing number of processors is used to solve it. A poorly scaling parallelization strategy will run well initially on a few dozen processors, and fail miserably if thousands of processors should be employed later.
- *Lack of algorithmic robustness*: Some parallelization strategies do not allow modifications to the kind of numerical and meteorological approaches adopted, impeding future progress in the scientific development of the field.
- *Necessity of massive rewriting of code*: Some parallelization strategies are deeply intertwined with the intrinsic communication geometry of numerical methods, and require low level recoding of numerical models. This results in substantial unnecessary coding work that may also make the code more fragile in the future.
- *Code dependence on machine architecture*: A parallelization strategy that too closely matches a particular parallel computer architecture, cannot be ported to

different successive parallel computer architectures without rewriting over and over again.

- *Semantics of parallel code that depend on the number of processors*: If the parallelization strategy allows a different number of processors to produce even minuscule differences in the output values, debugging of code becomes much more difficult over long periods of time, because bitwise reproducibility of results cannot be used as a debugging aid.

These concerns had to be carefully addressed when comparing alternative parallelization strategies. The goal was to come up with a strategy that was

- Scalable to a large number of processors
- Applicable to all foreseen algorithmic developments in all parts of the forecasting suite
- Well insulated from the code that implements the actual numerical methods and atmospheric physics
- Providing robust performance across a wide variety of foreseen parallel computer architectures
- Providing for bitwise reproducibility of results across all computer architectures and different numbers of processors

Following the road identified in the current research, these goals have been well attained in the operational forecasting context, both at ECMWF and at the Finnish Meteorological Institute. The more theoretical parts of the research pave the way for similar success in a number of future atmospheric applications.

## 5.1 Asymptotic analysis of parallel weather models

First, a top-down asymptotic analysis was made of the scalability of current numerical methods used in atmospheric models on massively parallel computers. Such an asymptotic analysis - limit performance as the number of processors approaches infinity - was motivated by two concerns:

- In the late 1980s, the most prominent massively parallel computers were getting massive indeed, with thousands of workstation type processors, or transputers, being embedded in a variety of interconnection networks.

- As resolution scales up, the sequential and parallel computational complexity of various algorithmic components of a forecast model scale differently. It is important to anticipate any potential complexity bottlenecks well in advance, so as to avoid spending code development effort in an eventual dead-end. It is also very important to remain as independent as possible of any particular computer type, in order to maximize the choice in future supercomputer purchases.

Asymptotic complexity analyses were developed to separate parallelisation issues that were dependent on the algorithm chosen from those that were inherent in the physical model used. Some basic algorithmic variants, particularly implicit equation solvers for both spectral and grid point models, were also coded in a parallel message passing style and compared in computational kernel benchmark tests. The results of these preliminary studies were published in the articles *Kauranne (1988-2, 1988-3, 1990), Barros and Kauranne (1990), Kauranne and Hoffmann (1990)* and the Licenciate thesis *Kauranne (1991).*

A general conclusion from the assessment was that weather models scale quite well, even up to thousands of processors, and that efficient parallel algorithms can be developed for both spectral and grid point models.

One of the novel techniques developed in the course of the above analysis was a generalisation of a Hockney-type dimensionless parallel efficiency measurement (*Hockney and Curington (1989)*) that makes it applicable to abstract algorithms just as well as to computers, thereby making asymptotic performance estimates of any differential equation based numerical model on any computer very straightforward to produce (*Kauranne and Hoffmann (1990)*). This asymptotic analysis makes both algorithms and parallel computers appear as abstract relativistic (i.e. hyperbolic) dynamical systems, with associated uncertainty principles and event horizons (*Kauranne (1991)*).

## 5.2 Identifying and benchmarking parallelization strategies

The second stage of the current research, following the optimistic conclusions of the above asymptotic assessment, was a quantitative study focused more closely on the methods, codes and model resolutions that had been decided upon on meteorological grounds. This involves taking into account also the lower order terms in asymptotic complexity analyses, as well as using finite size realistic models of parallel computers and model grids. Benchmarking was expanded to target increasingly complete parallel atmospheric models.

The latter half of the first stage and the first half of the second stage of current research were carried out while ECMWF was participating in a European parallel computing project called GENESIS that was funded from the Esprit programme of the European Commission. The article (**I**) summarizes the results of those studies, and serves as a more detailed introduction to the first stage described above, and the two subsequent articles in the current thesis (**II**) and (**III**).

Briefly, (**II**) makes a detailed quantitative scalability analysis of global spectral atmospheric models in the configuration they were run on vector supercomputers, on a range of hypothetical massively parallel computers, with a range of alternative parallelisation strategies. (**III**) implements a complete two-dimensional atmospheric model, employing the latest numerical schemes in operational use in the early 1990's, in a portable message passing programming paradigm and benchmarks it on a parallel computer.

### 5.2.1 Estimates of parallel scalability

The article (**II**) makes hypothetical implementations of spectral weather models on hypothetical parallel computers. We develop models that can forecast the sustained performance of spectral weather models when implemented on various hypothetical architectures and also on some real supercomputers.

A few key numbers are discussed in the article that give the simplest estimates for expected performance. They have proven to be very good, especially at identifying outright badly scaling high-performance computers.

The most important number for any proposed massively parallel high performance computer is its computational intensity, originally introduced in *Hockney and Curington (1989)*:

$$f_{1/2} = \frac{r_\infty}{b_\infty}$$

where $r_\infty$ is the sustained peak performance of every single processor multiplied by the total number of processors, and $b_\infty$ is the bisection bandwidth of the parallel computer. The bisection bandwidth is the total throughput in data that can be passed across any plane that divides the parallel computer in two chunks with an equal number of processors in each.

The computational intensity of a parallel computer is related to its expected sustained performance $r$ on a big application that uses all of its processors by the formula

$$r = \frac{r_\infty}{1 + f_{1/2}/f}$$

where $f$ is the computational intensity of the application. By the latter we mean the ratio of the number of results that need to be communicated across a bisection plane to the number of floating point operations carried out in unit time in the application.

A typical global spectral model has a computational intensity between 100 and 200, when the transposition strategy is used. If a parallel computer has a computational intensity that is of this same size or higher, the application will suffer a degradation of a factor of two or more in parallel efficiency. If the computational intensity is in the thousands, then more than 90 per cent of the wall clock time is spent in waiting for communication to complete.

The parallel vector supercomputers of the late 1990's had computational intensities between ten and 50. A Fujitsu VPP500 system discussed in (**II**) has a computational intensity of $f_{1/2} = 32$. This has meant a smaller than 50 per cent, more like 10 per cent, parallel overhead on parallel vector computers. This situation is now getting worse in the early 2000's with NUMA supercomputers. In the table below, a number of past, present and near future parallel supercomputers are compared in this respect:

| Computer | Processors | $r_\infty$ | $b_\infty$ | $f_{1/2}$ |
|---|---:|---:|---:|---:|
| | | (Gflop/s) | (Gword/s) | |
| Cray YMP | 8 | 2.7 | 4 | 0.667 |
| Cray C90 | 16 | 16 | 24 | 0.667 |
| nCUBE-2 | 8192 | 27 | 1.38 | 19.5 |
| Connection Machine CM-5 | 2048 | 262 | 0.64 | 409 |
| Intel Paragon XP/S | 4096 | 300 | 3.2 | 94 |
| Fujitsu VPP500 | 222 | 355 | 11.1 | 32.0 |
| Meiko CS-2 | 1024 | 410 | 102 | 4.0 |
| IBM SP 256 | 256 | 1098 (**75**) | 4 (**1**) | 275 (**75**) |
| IBM SP 704 | 704 | 3020 | 11 | 275 |
| Fujitsu HPC 2500 | 16 384 | 88 000 | 256 | 344 |
| Earth Simulator (NEC) | 5120 | 40 000 (**26 800**) | 975 | 41 (**27**) |

Table 5.1 - *Communication and computation performance characteristics of some past, present and future parallel supercomputers. The data are based on numbers published by the manufacturers. Bisection bandwidth has been computed for bidirectional communication with the largest communication capacity allowed by the architecture. The numbers in bold are based on computing the computational intensity from observed computation and communication values that have been reported in* **Dent et al (2002)** *for IBM SP and* **Shingu et al (2002)** *for the Earth Simulator.*

As an example, we shall look at IBM SP parallel supercomputers that have been acquired by both ECMWF and CSC in 2002. The first ECMWF system has 704 processors that each achieves roughly 1/8 of the sustained performance of a single Fujitsu VPP5000 vector processor according to measurements in *Dent et al (2002)* on a 256 processor system. This amounts to roughly 300 Mflop/s per processor, resulting in a likely peak performance of some 210 Gflop/s for the 704 processor system. The Colony switch that interconnects the processors achieves a bidirectional bisection bandwidth of 1 Gword/s on a 256 processor system. This means a 0.25 Gw/s bandwidth per each pair of communicating nodes of 32 CPUs. The architecture of the switch allows this number to scale up linearly to 11 pairs of simultaneously communicating clusters on a 704 processor system. Thus, the observed computational intensity of the ECMWF system would be $f_{1/2} = 75$. This is 2.5 times the computational intensity of the current Fujitsu VPP5000 vector system at ECMWF, while the theoretical peak computational intensity, at $f_{1/2} = 275$ is almost four times worse still.

A fine example of a system that scales well is the Earth Simulator. Constructed in 2001 - 2002, the Earth Simulator is currently the most powerful supercomputer in the world. It was built as a unique computational instrument, funded by the Japanese governement, to study the climate of the Earth. It comprises 5120 vector processors manufactured by NEC. These have been connected to one another by a crossbar switch built of thousands of kilometers of optical fibre, housed in a separate floor of the football field sized multifloor building that hosts the instrument. While such a system can hardly be adopted for mass production, it does guarantee the Earth Simulator ample communication bandwidth to carry out its task, unimpeded by poor parallel scalability.

### 5.2.2 Shallow water equations and spectral discretizations

This and the following subsection describe in mathematical terms the way global atmospheric models are formulated numerically, so as to give some background to the results of the current research effort, presented in the subsequent subsections. The sections 5.2.1 and 5.2.2 do not, as such, represent original research, although some of the points-of-view are new. These sections are based on the author's licenciate thesis (*Kauranne (1991)*), and the references therein.

The system of equations used in the benchmarks in (**III**) is one of the simplest models of the earth's atmosphere, the *shallow water equations*. It is a suitable benchmark case, since the *hydrostatic primitive equations* normally used in operational weather models can be viewed as a set of vertically stacked shallow water systems. The most important numerical analysis aspects in weather modelling pertain to horizontal motion, and are therefore present also in numerical methods for the solution of global spherical shallow water equations.

In a global model, it is necessary to state the equations in a coordinate system respecting sphericity. Normally, the spherical coordinates $(\lambda, \theta, z)$ are used. Here, $\lambda$ denotes longitude, $\theta$ denotes latitude and $z$ denotes height, i.e. radial distance from the earth's centre. The unit vectors in the corresponding directions are denoted by $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$. In advective form, the shallow water equations read:

$$\frac{d}{dt}\mathbf{v} = -f\hat{\mathbf{k}} \times \mathbf{v} - \nabla\Phi \tag{5.1}$$

$$\frac{d}{dt}\Phi = -\Phi\nabla \cdot \mathbf{v}, \tag{5.2}$$

where the total derivative $d/dt$ is defined in two-dimensions by

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla. \tag{5.3}$$

Above, $\mathbf{v} = (u, v)$ is the horizontal velocity vector and $f$ is the Coriolis parameter. The factor $f$ is given by $2\Omega \sin\theta$, where $\Omega$ is the angular velocity of the earth. The field $\Phi$ is the *geopotential* which is related to the height of the liquid surface (or depth of the liquid) $h$ by

$$\Phi = gh, \tag{5.4}$$

where $g$ is the acceleration by gravity. The horizontal gradient $\nabla$ is defined in spherical coordinates by

$$\nabla = \frac{\hat{\mathbf{i}}}{a\cos\theta}\frac{\partial}{\partial\lambda} + \frac{\hat{\mathbf{j}}}{a}\frac{\partial}{\partial\theta}, \tag{5.5}$$

where $a$ is the radius of the earth. The equation (5.1) describes conservation of momentum, and the equation (5.2) conservation of mass.

The most popular global discretization method, and indeed the one used in most operational global weather models at the moment, is the spectral method with spherical harmonics as the basis functions. In this method, all the fields are expanded in a series of spherical harmonics, which are eigenfunctions of the Laplacian on the sphere. They are tensor products of a longitudinal trigonometric polynomial with a latitudinal Legendre function. Hence, spherical harmonics are also eigenfunctions of longitudinal differentiation. It is also fairly straightforward to represent latitudinal differentiation on the spherical harmonic basis using the chain rule.

The two most attractive properties of the spherical harmonic basis are its sphericity - there are no geometrical problems with grid line convergence near the poles - and the ease of filtering aliased waves. Filtering is achieved by simply truncating the spectral series. The two dimensional series can be truncated in multiple ways, but it is most common to employ a triangular truncation at 2/3 of the number of latitudes $N$ used in the physical grid. This produces a spatially homogeneous and isotropic representation that also respects spherical symmetries: the pole can be rotated anywhere and the basis still spans the same function space. "Triangularity" of the set of spectral coefficients emerges naturally, since spherical harmonics are only defined for those longitudinal-latitudinal index pairs $(m, n)$ for which $n > m$. The quadratic nonlinear terms in (5.1) and (5.2) are always computed on the full grid in grid space. It can be proved that truncation to 2/3 in spectral space eliminates the aliasing due to quadratic nonlinearities.

Projecting scalar fields onto the spherical harmonic basis is carried out by doing a *spectral transformation* to the spectral space where grid functions are represented by their Galerkin coefficients with respect to the basis of spherical harmonics. Time stepping is done in the spectral space, and a back transformation projects the new field back to the space of grid functions. Employing the triangular truncation, a two dimensional scalar field $u(\lambda,\theta)$, defined in spherical coordinates, is expanded as:

$$u(\lambda,\theta)=\sum_{m=-M}^{M}\sum_{n=|m|}^{M}\hat{u}_n^m Y_n^m(\lambda,\theta) \quad , \tag{5.6}$$

where $Y_n^m(\lambda,\theta)=e^{im\lambda}P_n^m(\theta)$ are the spherical harmonics, $\lambda$ represents the longitude, $\theta$ the latitude and $P_n^m$ is the Legendre polynomial of degree $n$ and order $m$.

The first step in the spectral transformation process is the evaluation of the spectral Galerkin coefficients of the field $u(\lambda,\theta)$ defined by

$$\hat{u}_n^m=\frac{1}{4\pi}\int_0^{2\pi}\int_{-1}^{1}u(\lambda,\mu)\bar{Y}_n^m(\lambda,\mu)d\mu d\lambda \quad , \tag{5.7}$$

where $\mu=\sin\theta$, and the overbar denotes complex conjugation. This is done numerically in two steps, with the aid of a Gauss-Legendre quadrature which is computed on a Gaussian grid $\{(\lambda_i,\mu_j),\ i=0,\ldots,2N-1,\ j=1,\ldots,N\}$, where the $\lambda_i$'s are uniformly spaced and the $\mu_j$'s are the roots of the Legendre polynomial $P_N^0(\mu)$. With the 2/3-truncation, $M\approx 2N/3$. First, the direct Fourier transforms are computed by

$$u^m(\mu_j)=\frac{1}{2N}\sum_{l=0}^{2N-1}u(\lambda_l,\mu_j)e^{-im\lambda_l} \tag{5.8}$$

for all $j$ and $m$ and then the direct Legendre transforms by

$$\hat{u}_n^m=\sum_{j=1}^{N}w(\mu_j)u^m(\mu_j)P_n^m(\mu_j) \tag{5.9}$$

for all $-M\leq m\leq M$ and $|m|\leq n\leq M$, where $w(\mu_j)$ are the Gauss-Legendre quadrature weights. After the time stepping, the truncated back transform on the Gaussian grid is obtained by computing the inverse Legendre transforms by

$$u_m(\mu_j)=\sum_{n=|m|}^{M}\hat{u}_n^m P_n^m(\mu_j) \tag{5.10}$$

for all $j$ and $m$ followed by the inverse Fourier transforms

$$u(\lambda_l,\mu_j)=\sum_{m=-M}^{M}u_m(\mu_j)e^{im\lambda_l} \quad . \tag{5.11}$$

The whole transformation process consists of performing (Fast) Fourier transforms in steps (5.8) and (5.11) and discrete Legendre transforms in steps (5.9) and (5.10).

### 5.2.3 Time stepping schemes

The main computational concern in time stepping schemes for partial differential equations is their numerical stability. Most of the schemes used in atmospheric models are leap-frog type two time-level, first or second order schemes. Memory limitations on vector supercomputers have in the past precluded storing several time slices of the solution in the memory. The stability of two-level schemes is governed by the Courant-Friedrichs-Lewy (CFL) stability condition (*Courant et al (1928)*). It maintains that for any wave or motion present in the discrete equations, the inequality

$$\frac{|\mathbf{u}|\Delta t}{\Delta x} \leq 1 \tag{5.12}$$

must hold uniformly. Here $|\mathbf{u}|$ is the wave phase speed, $\Delta t$ is the time step and $\Delta x$ is the spatial grid length.

When the spatial resolution is refined, the CFL condition enforces a proportional linear cut in the time step, unless some of the fastest waves can be removed from the final field updating stage in the time stepping scheme. This can be achieved by moving the terms responsible for these waves into the left hand side of the time stepping scheme, implying implicit treatment. Implicit treatment retains all types of waves present in the initial state or generated as a response to forcing, but it prevents their numerical amplification in the field updating stage, which is necessarily explicit.

The most common *semi-implicit* time stepping scheme does a splitting of dynamic terms in the system of equations to a rapidly oscillating linear part and a slowly varying nonlinear part. Instead of using a short time step dictated by a CFL condition for fast, short waves, the terms responsible for sound and linear inertia-gravity waves, the two fastest, linear atmospheric wave types, are solved for implicitly. This is achieved by projecting the solution fields onto the kernel of the implicitly treated part of the operator, so that there is no numerical amplification of sound or linear inertia-gravity waves.

The drawback of filtering sound waves in a full three-dimensional atmospheric model is that the primitive equations are no longer isotropically hyperbolic, unlike the shallow water and Euler equations. The set of solutions to an elliptically constrained hyperbolic system is not dense in any Hilbert space of suitably smooth functions. Instead, it is an infinite dimensional submanifold defined by the elliptic relation maintaining hydrostaticity, incompressibility or anelasticity, which are various assumptions, one of which needs to be made in order to avoid very strong constraints on the time step by vertical waves. Since the fluid is assumed inviscid, this submanifold is not an attractor, implying that the problem is ill-posed. In numerical algorithms, this is corrected by artificial viscosity or by projecting the solution onto the submanifold by solving an elliptic equation every time step in a semi-implicit time stepping scheme. These issues are discussed in *Kauranne (1991)*.

The cost of semi-implicit time stepping varies with the strength of the balance assumption. Assuming only incompressibility renders the elliptic equation to be solved spatially separable. The vertical layers of the model can then be decoupled with an explicit eigendecomposition in the vertical, leaving a set of two dimensional *Helmholtz equations* to be solved:

$$\nabla^2\psi + \alpha_k\psi = \zeta \tag{5.13}$$

where $\psi$ is the stream function to be solved for, $\zeta$ is the vorticity and $\alpha_k$ is the $k$th separation constant, i.e. the eigenvalue corresponding to the $k$th vertical eigenmode. A similar Helmholtz equation also holds between the velocity potential $\chi$ and the divergence $D$. Selecting one field from the pair $(\psi, \zeta)$ and another from the pair $(\chi, D)$ provides alternative coordinate pairs for the two dimensional wind field $(u, v)$. The ECMWF spectral model uses a $(\zeta, D)$-basis as the primary representation of the wind field.

When using a spectral method with spherical harmonics, the relation

$$-\Delta Y_n^m + \alpha_k Y_n^m = (\alpha_k + n(n+1))Y_n^m \tag{5.14}$$

states the diagonalizability of the Helmholtz operator on spherical harmonics. (5.14) enables, once the right-hand-side expansion

$$\zeta(\lambda, \theta) = \sum_{m=-M}^{M} \sum_{n=|m|}^{M} \hat{\zeta}_n^m Y_n^m(\lambda, \theta) \tag{5.15}$$

is known, the direct computation of the spectral coefficients of $\psi$ by dividing the spectral transform of the data $\hat{\zeta}_n^m$ by the corresponding eigenvalue of the Helmholtz operator $\alpha_k + n(n+1)$:

$$\hat{\psi}_n^m = \hat{\zeta}_n^m / (\alpha_k + n(n+1)) \quad . \tag{5.16}$$

The corresponding three dimensional stream function or velocity potential field is calculated by adding the appropriate contributions from each of the vertical eigenmodes. This is carried out through a matrix multiplication by a full $K \times K$ matrix, where $K$ is the number of vertical layers, to get the values for each vertical column.

The next limiting velocity, after sound waves and a dominant linear part of inertia-gravity waves have been eliminated, is advection, i.e. wind speed. Because this involves the nonlinear momentum transport terms, a fully implicit treatment of advection terms would amount to solving a large, nonlinear system every time step. This is prohibitively expensive. Instead, the advection terms can also be split into a basic flow and a perturbation flow. In *semi-Lagrangian* advection techniques, the basic flow is taken to be the wind field of the previous time step, possibly interpolated to higher order accuracy by iteration. The fields advected have to be interpolated to an even higher order than the advection terms, in order not to introduce a new dominant discretization error term. The limiting wave speed in the CFL condition in semi-Lagrangian advection is the temporal variation of the wind field multiplied by the length of the time step.

## 5.3 The transposition strategy

The most significant innovation of the current research, extended and applied in both papers (**II**) and (**III**), and first presented in a two-dimensional form in *Barros and Kauranne (1990)*, is a general parallelization strategy for implicitly solved time-dependent dynamical systems with a discrete basis that is topologically rectilinear. It is called the *Transposition Strategy.* Analogous parallelization strategies have been used on shared memory computers, but we believe ourselves to be among the first independent research teams that chose to adopt this quite unorthodox strategy for an operational model on a distributed memory parallel computer.

### 5.3.1 An overview of parallelization strategies for partial differential equation based numerical models

Partial differential equations are the natural model for most continuous processes in physics. When looking at models of classical physics, they feature most often two or three spatial dimensions. The spatial dimensions are complemented with the time dimension, when we attempt to model a transient phenomenon, such as the weather.

Nature works in parallel. Molecules are assumed to feel only their immediate surroundings, when they collide with neighbouring molecules. There are many possible levels to reproduce this parallelism in numerical models based on partial differential equations. One, albeit nonexhaustive, way to classify them has been presented in *Kauranne (1988-2)* as follows:

1. Job level parallelism (operating system)
2. Task level parallelism (macrotasking, subroutine or block level)
3. Code level parallelism (microtasking, parallelizing compilers, SIMD-machines, dataflow machines)
4. Linear algebraic parallelism (parallel vector and matrix algorithm libraries)
5. PDE-numerical parallelism (parallelizing over a set of discrete basis functions)

6. PDE-theoretical parallelism (“microlocal” parallelism over localized functions on the cotangent bundle, rate of decay of Green’s functions)

7. Physical parallelism (physical relations between meteorological fields, Monte Carlo molecular dynamics in the extreme)

The levels of specific interest to parallelizing partial differential equation models are four, five and six. Linear algebraic parallelism is applicable if the numerical method employed in solving the system of partial differential equations has been cast in a generic matrix form. This works for most numerical methods for linear problems, and spatially regular discretizations.

Unfortunately, modern primitive equation based numerical weather models are neither. The primitive equations, being a special case of the Navier-Stokes equations, are nonlinear. Modern solvers use specific advection techniques, such as semi-Lagrangian advection, that change their communication pattern spatially every time step. The grids used in global and limited area models are also often irregular, in the interest of computational economy, and because of the nonexistence of fully regular tilings of the sphere.

Even the geometrically regular parts of the time stepping scheme, such as semi-implicit time stepping in spectral space, are carried out by special fast solvers of multigrid or Fast Fourier Transform type. The only part of time stepping that is readily amenable to linear algebraic processing is the transformation to vertical eigenmodes. Here, however, the vectors are relatively short, up to a hundred elements, so that parallelizing individual such operations across many processors is not attractive, because of the overheads caused by it. It is far more expedient to parallelize across many such operations.

As to parallelizing across discrete basis functions, this is the prevailing approach to parallelization in most partial differential equation solvers. Such a parallelization strategy is called *Domain Decomposition*, and it involves allocating a fixed spatial domain to every processor. As the discrete basis functions are spatially local, their supports are overlapping only with a fixed number of neighbouring basis functions, irrespective of model size. Information is exchanged only along the boundaries of such domains.

Domain decomposition is the natural parallelization strategy for explicit time stepping. It also simulates the parallelism in nature. It has a very attractive scaling behaviour for explicit time stepping schemes, because the width of the ”halo” accross which messages need to be exchanged between neighbouring processors, stays constant, even if the size of the subgrid assigned to each processor may increase with increasing model size. This results in an improving computation over communication ratio with increasing model size on a fixed number of processors.

However, in global atmospheric models, explicit time stepping is prohibitively expensive. Because of the Courant-Friedrichs-Lewy condition, the time step necessitated at grid points near the poles is hundreds of times shorter than the one allowed by semi-implicit, semi-Lagrangian time stepping. This means that the advantage in parallel efficiency provided by domain decomposition will be more than lost in a massive unnecessary increase in the serial complexity of the underlying algorithm. This is true for both spectral and grid point models.

We therefore have to look for optimal parallelization strategies for semi-implicit time stepping schemes. Adopting even a partially implicit approach in time stepping, especially if it features all three spatial dimensions - as it must in global weather models - completely changes the nature of data communication patterns on a parallel computer. Whereas it was necessary to exchange information only along subdomain boundaries in domain decomposition, in implicit time stepping, every grid point is influenced by every other grid point every time step. This is a mathematical fact that emerges from the elliptic nature of the implicitly treated part of the partial differential operator at hand, and there is no way to avoid it.

In any static domain decomposition, information must be collected from all other subdomains, too. In the way that the spectral numerical algorithms employed in global weather models go, this information is collected along each coordinate axis at a time in a spectral transform. The vertical direction is handled by direct matrix multiplication along the vertical eigenmodes. In the longitudinal direction, a Fast Fourier Transform is applied. In the latitudinal direction, a Legendre transform is carried out, again by matrix multiplication. Analogous communication patterns are present also in semi-implicit grid point models.

Since we do not wish to change these underlying numerical methods, a static domain decomposition implies communicating data by essentially operating a pipeline that transmits all the subdomains, one subdomain per stage, through all the processors that have been assigned to a subdomain on the same "belt", and in at least two directions out of the three. An optimal shape of such a belt produces roughly square shaped subdomains, because in strongly elongated subdomains, the entire data structure ends up being transmitted through every processor every time step. This results in a total data communication volume of $pn^3$ per time step, where $p$ is the number of processors and $n$ is the grid size in one dimension.

Even in square subdomains, the volume of data to be communicated is proportional to the size of the entire data structure, divided by just the square root of the number of processors. The total volume of data communicated every time step is therefore $\sqrt{p}n^3$, because there are $\sqrt{p}$ stages in every such pipeline.

The transposition strategy goes against the common wisdom in parallel computing that calls for maximizing locality in computations, and which therefore suggests static domain decomposition based parallelization strategies. In the transposition strategy, global three-dimensional fields are completely reorganized across all the processors several times every time step. Transposition therefore entails a dynamic domain decomposition that changes several times every time step.

In transposition strategy, all the computations are carried out locally within each processor. The numerical results produced are therefore bitwise identical on any number of processors. Between computation steps, data is communicated so, that all the necessary inputs reside in the very processors where they will be needed to perform the next step of computations. This ensures that computation and communication steps in the code are almost completely separated from one another. Communication constructs are hidden in a separate subroutine library, and numerical methods can be modified without this interfering with the parallel efficiency of the code.

Because of the rectilinear structure of the transforms employed in spectral and grid point based weather models alike, all communications proceed along a single direction at a time. Transposition strategy maximizes the parallelism at each such stage by doing a roughly square shaped domain decomposition along the two directions that are not involved in each computation step at a time. As a result, there is no need to communicate along subdomain boundaries at all.

The price of this benefit is the necessity to shuffle the entire data structure altogether across the parallel computer six times every time step. But when seen from a global communication point of view, the volume of communication involved in this is just $6n^3$ per time step. Unlike in domain decomposition, this does not depend on the number of processors. The asymptotic parallelism analysis carried out in **(II)** clearly indicated that global communication bandwidth is the worst bottleneck on massively parallel computers. With respect to this, transposition strategy attains optimal scalability of the volume of global communication, up to at most a constant factor.

In both estimates and simulations, the transposition strategy appears no less efficient on realistic massively parallel computers than the best alternative static domain decomposition based parallelization strategy. The geometric idea of transposition is illustrated in the pictures in **(III)**, and results of comparison benchmarks are reported in *Foster and Worley (1994).* In that reference, the authors also develop an elaborate communication strategy for domain decomposition that attains the same asymptotic efficiency as the transposition strategy. However, this strategy requires a lot more modifications to the code implementing the numerical methods, as it must be implemented at a level very close to the numerics.

The existence of such a strategy, as well as the similar communicative behaviour of both spectral and grid point models, demonstrates, however, that the communication volumes and patterns involved in solving partial different equations on parallel computers are indeed very much like natural constraints, similar in spirit to the Heisenberg uncertainty principle. An asymptotic study of the elliptic and hyperbolic communication patterns implied by different parallelization strategies has been carried out in the Licenciate Thesis *Kauranne (1991).*

### 5.3.2 Transposition strategy in primitive equation models

As can be seen from the composition of a spectrally represented field by spectral transformations, as in (5.8) and (5.9), as well as for the inverse transformations in (5.10) and (5.11), a spherical spectral discretization is naturally separable into successive latitudinal and longitudinal stages, where all the data dependencies in each are one-dimensional only. The equation (5.15) shows why a complete set of spectral transforms are needed every time step, if we are to retain a reasonably long time step by using semi-implicit time stepping.

In static domain decomposition strategies, a rectangular horizontal block is statically allocated to each processor. This makes it necessary to split all the sums in (5.8), (5.9), (5.10) and (5.11) into two-stage summing loops where the length of subsums depends on the number of processors. Such an approach makes it necessary to parallelize the computational kernel. It also makes the sums dependent on the number of processors, since rounding errors accumulate differently, depending on the subsum pattern. While this is not serious from the numerical stability point-of-view, it makes debugging parallel code a lot harder, since minuscule deviations in results do not always signal a bug in the code.

In the transposition strategy, the key idea is to reorganize data fields across processors before computing sums, so that all the data needed for computing each sum reside on the processor that is to carry out the summing, at the time when the summing is due to commence. The sums in (5.8), (5.9), (5.10) and (5.11) are therefore not altered at all. Instead, all the fields needed to be transformed are reshuffled between the orthogonal summing stages. In a two dimensional problem, this results in a horizontal transposition of the fields over a processor grid - hence the name transposition strategy.

Furthermore, in the case of full three-dimensional weather models, the transposition strategy can be extended to three dimensions. As is described in the second last paragraph of subsection 5.2.2, the vertical part of the semi-implicit time stepping scheme amounts to a vertical matrix multiplication. The matrices involved are dense but relatively small, and incur data dependencies only in the vertical direction. Therefore another transposition - with respect to a different axis - makes the data available for an unaltered vertical matrix multiplication on a single processor. The very same pattern is followed in the so-called physics computations between the dynamic time stepping steps. In physics computations, various algebraically approximated small scale atmospheric processes are computed. Virtually all of these incur data dependencies only between successive atmospheric layers - hence in the vertical direction only, please see Figure 2 in (**III**).

The transposition strategy provides the additional benefit of simple coding - particularly for spectral models, since it avoids parallelizing any of the computational kernels in a numerical scheme - and the best scaling behaviour on large numbers of processors. The transposition strategy is the parallelization strategy that in the asymptotic limit has the smallest data volume to communicate of all parallelization strategies for any implicit time stepping scheme, for spectral and grid point models alike, as explained in subsection 5.3.1 above.

Since the current research, a thorough analysis and a careful parameterized implementation has been made of the two principal families of parallelization strategies for global atmospheric models - transposition versus static domain decomposition based - and, while finding various parallel computers on which each of the numerous versions and combinations of the strategies belonging to each family proves to be the most efficient, the authors find the transposition strategy to be a robust choice on virtually all current computers (*Foster and Worley (1994), Worley, Foster and Toonen (1994)*).

Model benchmarking tests initiated in (**III**) were expanded to a full two-dimensional version of the IFS model in *Gärtel et al. (1994-1)*, see also *Gärtel et al. (1994-2)*, and eventually to the operational version of IFS (*Barros et al. (1994), Dent et al. (1994)*). On the basis of the excellent results obtained in these benchmarks, the transposition strategy has become the parallelization strategy of choice for IFS, the first operational global weather model that has operated on a massively parallel, distributed memory computer ever since 1996.

## 5.4 Theoretical investigation of coupled parallel methods for data assimilation and ensemble forecasting

Simultaneously with the second stage above, qualitative asymptotic analysis was expanded to cover variational data assimilation and ensemble forecasting. The article (**IV**) reports research initially undertaken as a safeguard against possible inefficiency in parallelizing low resolution model runs that are necessary in both variational data assimilation and ensemble forecasting, over a large number of processors.

### 5.4.1 Basics of variational data assimilation

Data assimilation is the process by which a weather model takes into account the information contained in weather observations around the world since the previous forecast was run. Its result is called the *analysis*, and it is used as an initial state when the subsequent forecast is run. The most important observation is the previous forecast itself and it is called the observation background in this context.

Earlier on, data assimilation was carried out by purely statistical means. Essentially, it amounted to a least squares average of the observations and the background, weighted by the inverse of their measured observation variances. Each observation was leveled out by radial basis functions in space, so as to fill out the entire space with an initial state for a subsequent forecast. Such an assimilation scheme is called Optimum Interpolation.

Since the 1980's, it has become apparent that substantial improvements in forecast quality can be obtained by taking the weather pattern into account, when interpreting local weather observations. After all, most observations have their significance in an area where they are transported by the wind. Such a time dependent assimilation process can be defined via optimal control theory, and it is called variation assimilation, because we try and minimize the first variation of a least squares cost functional that measures the deviation between the analysis and the observations.

Let the evolution of the atmosphere be governed by the nonlinear partial differential equation system

$$\frac{\partial x(t)}{\partial t} = F(x)$$

$$x(t_0) = u$$

The vector valued atmospheric initial state $x(t_0)$ is typically a six component vector field, defined on a spherical shell that contains the earth's atmosphere up to a height of 70 km or more. The vector $x(t)$ is the trajectory emanating from this initial state to up to ten days into the future. The components of the vector field are most often the three components of the wind, temperature, pressure and liquid water content. Five of the component fields are assumed smooth, whereas liquid water may be a discontinuous field.

We define the time evolution of the model state $x(t)$ from an initial state $x(t_0)$ to a later state $x(t_n)$, using the model's resolvent operator $\mathcal{M}(t_n, t_0)$, by

$$x(t_n) = \mathcal{M}(t_n, t_0)(x(t_0))$$

Linearizing around a trajectory $\overline{x}(t)$, started from the basic state $\overline{x}(t_0) = u$, we can describe the first order evolution of a small perturbation $\delta x(t), \quad \delta x(t_0) = \delta u$ to the basic state by

$$\frac{\partial(\overline{x}(t) + \delta x(t))}{\partial t} = F(\overline{x}(t) + \delta x(t))$$

$$= F(\overline{x}(t)) + \frac{\partial F}{\partial \overline{x}} \delta x(t) + \frac{1}{2} \delta x^* \frac{\partial^2 F}{\partial \overline{x}^2} \delta x + \mathcal{O}(||\delta x||^3), \tag{5.17}$$

$$\overline{x}(t_0) + \delta x(t_0) = u + \delta u,$$

where $^*$ denotes Hermitian transpose and $|| \cdot ||$ the Euclidean norm. Truncating the above variational equation to first order in the state variable $x$, we obtain the *tangent linear model equations* for the small perturbation $\delta x$:

$$\frac{\partial \delta x(t)}{\partial t} = \frac{\partial F}{\partial \overline{x}} \delta x(t)$$

$$\delta x(t_0) = \delta u.$$

We shall denote by $\mathcal{R}_{\overline{x}}(t_n, t_0)$ the resolvent of the tangent linear model, linearized around $\overline{x}(t)$. The tangent linear equations then assume the form

$$\delta x(t_n) = \mathcal{R}_{\overline{x}}(t_n, t_0)(\delta x(t_0)).$$

Defining a least squares cost functional $J(u)$ with the associated inner product to measure the weighted distance between the trajectory $x(t)$ that emanates from the initial state $u$ and a set of retrieved observations and a given background forecast, jointly given through their weighted linear combination $y(t)$, we can formulate 4D variational assimilation as the minimization problem for the cost functional $J(u)$:

$$J(u) = \frac{1}{2} \int_{t_0}^{t_n} (\mathcal{H}(\mathcal{M}(t, t_0)(x(t_0)) - y(t))^T Q^{-1} (\mathcal{H}(\mathcal{M}(t, t_0)(x(t_0)) - y(t)) dt. \tag{5.18}$$

In the formula above, $\mathcal{H}$ is a spatial, possibly nonlinear, observation operator, typically incorporating a projection onto a finite dimensional subspace, that translates the model state $x(t)$ to an observation $y(t)$. Associated with it, $\mathcal{H}$ has a tangent linear operator $\mathcal{H}'$ and its adjoint $\mathcal{H}'^*$. $Q$ is a (normally, but not necessarily, time and state independent) symmetric and positive definite estimate of the error covariance matrix of the first guess and of the observations, acting on the model state variables. Below, $Q^{-1}$ is assumed to define the scalar product used in the minimization and denoted by $< \cdot, \cdot >_{Q^{-1}}$. $< \cdot, \cdot >$ will be used to denote the Euclidean inner product.

Assuming $J(u)$ to be continuously Fréchet differentiable, the minimization is solved by an $x$ satisfying the nonlinear equation:

$$\nabla_u J = \int_{t_0}^{t_n} \mathcal{R}^*(t,t_0)\mathcal{H}'^* Q^{-1}(\mathcal{H}(\mathcal{M}(t,t_0)(x(t_0)) - y(t))dt = 0, \tag{5.19}$$

where $\mathcal{R}^*(t,t_0)$ denotes the adjoint of the resolvent of the tangent linear model $\mathcal{R}(t,t_0)$.

We note that for solving the variational data assimilation problem, it is not necessary to assume model dynamics to be linearizable as, for example, *Le Dimet and Navon (1996)* have shown. Full model dynamics are, in fact, normally not linearizable, because the small scale nonlinear effects create very sharp features that have to be modelled as discontinuous processes. Cloud entrainment and detrainment are good examples.

However, since in this study we discuss primarily large scale atmospheric motion over a period of a few days, the dynamics are treated as linearizable. This allows us to use a singular value decomposition, which will prove very helpful a tool in understanding the computational behaviour of variational assimilation and ensemble forecasting methods.

Since the operators $\mathcal{R}^*$, $\mathcal{H}'^*$ and $Q^{-1}$ are linear, (5.19) is equivalent to solving the inhomogeneous equation

$$\int_{t_0}^{t_n} \mathcal{R}^*(t,t_0)\mathcal{H}'^* Q^{-1}\mathcal{H}(\mathcal{M}(t,t_0)(x(t_0))dt = \int_{t_0}^{t_n} \mathcal{R}^*(t,t_0)\mathcal{H}'^* Q^{-1} y(t)dt \tag{5.20}$$

with the nonlinear operator $\int_{t_0}^{t_n} \mathcal{R}^*\mathcal{H}'^* Q^{-1}\mathcal{H}\mathcal{M}dt$. The optimal initial state $u_0$, called the *analysis* is obtained as the solution to the equation (5.20). The search for the analysis is started at any point of the process from a guess $u = u_0 + \delta u$. The vector $-\delta u$ is termed the analysis increment.

In operational implementation, 4D variational assimilation over a time interval $[t_0, t_1]$ of, say, 24 hours thus consists of the following components:

1. A nonlinear forward model $\mathcal{M}(t_1, t_0)$ that integrates an initial state $x(t_0)$ to a final state $x(t_1)$.

2. The corresponding tangent linear model $\mathcal{R}(t_1, t_0)$, linearized around a trajectory computed with the nonlinear model.

3. The adjoint model $\mathcal{R}^*(t_0, t_1)$ of the tangent linear model, integrating the final dual state $\delta' x(t_1)$ to the initial dual state $\delta' x(t_0)$ backwards in time.

4. A spatial, possibly nonlinear, observation operator $\mathcal{H}$, that translates the model state $x(t)$ to an observation $y(t)$, together with its tangent linear operator $\mathcal{H}'$ and its adjoint $\mathcal{H}'^*$.

5. A spatial observation error covariance matrix $Q$.

6. A least squares cost function

$$J(u) = \frac{1}{2}\int_{t_0}^{t_n} (\mathcal{H}(\mathcal{M}(t,t_0)(u)) - y(t))^T Q^{-1} (\mathcal{H}(\mathcal{M}(t,t_0)(u)) - y(t))dt$$

   on the space of model initial states at time $t_0$, that measures the deviation of the observations $y(t)$ from the corresponding 'quasi-observations', derived with the observation operator $\mathcal{H}$ from the model state $x(t)$, integrated from the initial state $u$ with the model $\mathcal{M}(t,t_0)$.

4DVAR is an iterative procedure to minimize $J(u)$ over all initial states $u$. The minimization is accomplished by gradient based minimization algorithms, such as Limited Memory Quasi-Newton (LMQN), or Conjugate Gradient (CG) methods.

The gradient $\nabla_u J(u)$ of the cost function at an initial state $u$ is

$$\nabla_u J(u) = \int_{t_0}^{t_n} \mathcal{R}^*(t_0,t)\mathcal{H}'^* Q^{-1}(\mathcal{H}(\mathcal{M}(t,t_0)(u)) - y(t))dt$$

The minimization consists of the following steps:

1. Given an initial state $u^i$, integrate the nonlinear model over the assimilation period, storing the entire model state for every time step and the deviations from the observations for all the observations.

2. Starting from a homogeneous final condition, integrate the adjoint model backwards in time over the assimilation period, using the stored trajectory as coefficients and the stored deviations as forcing, each applied at the time of the corresponding observation.

3. The resulting initial dual state is the gradient of the cost function. Using that, the minimization algorithm determines a new search direction and performs a line search along it to arrive at a correction $s^i$ to the initial state $u^i$.

4. Assign $u^{i+1} = u^i + s^i$ and repeat the process until $||\nabla_u J(u)|| < \epsilon$, for a given stopping tolerance $\epsilon$.

### 5.4.2 Basics of ensemble forecasting

Ensemble forecasting is a method to address the variability of weather forecasts from day to day. Even if we produce superbly accurate individual weather forecasts, the chaotic nature of atmospheric motion causes the forecasts at any one location to be very sensitive to even small errors in observations, or model resolution.

It has been deemed useful therefore to provide, if possible, some indication along the weather forecast as to whether the forecasters believe it to be reliable. The reliability of forecasts depends on the sensitivity of the atmosphere at a particular location at a particular time. In the middle of a large high pressure area, the weather is stable. Close to a cyclone or hurricane track, it is very unstable. In general, so called baroclinic phenomena, such as mid-latitude cyclones, cause the atmosphere to undergo rapid and dramatic changes.

Baroclinic features are a permanent character in global weather patterns, since they are the most important mechanism by which the atmosphere conveys heat from the tropics to the extratropics. Large baroclinic features show up as the singular functions of a local linearization of the primitive equations that correspond to large singular values, i.e. rapidly growing weather patterns.

In ensemble forecasting, instead of a single forecast, a bunch of related forecasts is produced. Then the statistical spread of such a bunch around any particular location is used as a criterion towards the sensitivity, or lack of it, of the weather pattern there, and thereby also towards the reliability, or lack of it, of the prevailing forecast. In first attempts, the bunch - the ensemble - of forecasts was produced by perturbing the prevailing analysis in random directions by a vector. The size of the vector was chosen to correspond to the estimated uncertainty in observations.

It turned out, however, that the space of potentially explosive atmospheric patterns is very high dimensional. A prohibitively large number of perturbed forecasts were needed to span it. It then became mandatory to try and find more efficient ways to choose the initial perturbations to the analysis. A good approach was to take the tangent linear model of the forecast model, and take the singular vectors corresponding to its largest singular values as the perturbation directions. These singular values and singular vectors can be computed with standard Krylov space methods for symmetric matrices, in particular by the Lanczos algorithm.

Several recent studies have indicated that over assimilation periods less than 48 hours long, the dynamics of the atmosphere are predominantly linear (*Courtier and Talagrand (1987)*, *Lacarra and Talagrand (1988)*, *Vukicevic (1991)*). It also appears that the identification of the linearly most unstable modes over short periods of integration are valuable to the problem of flow predictability over longer periods (*Molteni and*

*Palmer (1991)* and *Mureau, Molteni and Palmer (1991)*).

Under the assumption of near linearity, the operator $\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H} \mathcal{M} dt$ is well approximated by its linear component $\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R} dt$. In this case, as shown in *Rabier and Courtier (1992)*, $\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R} dt$ is a good approximation to the Hessian matrix $J''$ at the minimum, while its inverse $(\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R} dt)^{-1}$ is a good approximation to the model state error covariance matrix $(J'')^{-1}$ at the minimum $u_0$. This assumption of near linearity is crucial to the validity of the algorithmic similarities to be discussed below.

Because the cost functional $J(u)$ was assumed differentiable, its Hessian will always be well approximated by its linear component in some neighbourhood of the minimum. During the last, local stages of the minimization, the descent algorithm will therefore always approximately be solving the equation

$$\int_{t_0}^{t_n} \mathcal{R}^*(t,t_0) \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R}(t,t_0) x(t_0) dt = \int_{t_0}^{t_n} \mathcal{R}^*(t,t_0) \mathcal{H}'^* Q^{-1} y(t) dt \qquad (5.21)$$

with the linear operator $\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R} dt$, even though globally it will be solving the nonlinear equation (5.20).

The singular values of the tangent linear model $\mathcal{R}$ over the time interval $[t_0, t_n]$, modified by the inverse of the observation error covariance matrix $Q$, are precisely the eigenvalues of the operator $\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R} dt$ in (5.21). The singular vectors corresponding to the largest singular values are the linearly most unstable modes over the period $[t_0, t_n]$. They appear to be good perturbation directions for an ensuing ensemble forecast (*Mureau, Molteni and Palmer (1991)*), especially if they could be computed from the operational model itself and 'on-line' for the current state of the atmosphere.

If the descent algorithm employed in 4D variational assimilation would simultaneously be able to provide good approximations to the eigenvalues and eigenvectors of $\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R} dt$, these could be used as perturbation directions in an ensuing ensemble forecast. Lanczos type optimization methods indeed produce such good approximations, as will be demonstrated below.

As is readily seen, a necessary and sufficient condition for an atmospheric state $x(t_0)$ to be an eigenvector of $\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R} dt$ is that it satisfies the linear equation

$$\int_{t_0}^{t_n} \mathcal{R}^*(t,t_0)) \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R}(t,t_0) x(t_0) dt = \lambda x(t_0) \qquad (5.22)$$

for some positive real number $\lambda$.

Eigenvalues of the symmetric, positive definite operator $\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R} dt$ are themselves extremal values of a quadratic cost functional, namely the *Rayleigh quotient* $R(u)$:

$$R(u) = \frac{< x(t_0), \int_{t_0}^{t_n} \mathcal{R}^*(t,t_0))\mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R}(t,t_0) x(t_0) dt >}{< x(t_0), x(t_0) >} \tag{5.23}$$

Under the assumption of near linearity of atmospheric dynamics, both solving the eigenproblem and solving the 4D variational assimilation problem therefore results in a linear equation - (5.22) or (5.21), respectively - with the operator $\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R} dt$.

Lanczos type methods for any symmetric, positive definite linear operator $A$ are iterative methods that approximate the solution to the equation $Ax = b$ at the $k$'th iteration in the $k$'th *Krylov space* $K^k(A, b)$, generated by the operator $A$ and a right hand side vector $b$. The $k$'th Krylov space is defined as the $(k+1)$-dimensional vector space spanned by the basis

$$\{b, Ab, A^2b, ..., A^kb\}$$

Each Lanczos type method defines a Krylov space approximation to the equation that is the best one with respect to some norm. Different Lanczos methods differ mainly in the selection of this norm.

Equivalently, Krylov spaces can be defined as the appropriate spaces to define best approximations to the operator $A^{-1}$ by $k$'th degree (operator valued) polynomials of $A$. A particular polynomial that reflects the goodness of an iterative method is its *residual polynomial* $R_k(A)$ that satisfies the equation:

$$r^{(k)} = R_k(A) r^{(0)}$$

where $r^{(k)}$ is the $k$'th residual and $r^{(0)} = b - Ax^{(0)}$, $x^{(0)}$ being the first guess. In the case of a symmetric, positive definite operator $A$, the above approximation problem is isomorphic, after scaling by a constant, to a best approximation problem of the real zeros of the operator valued analytic function of a complex argument $z$:

$$\psi(z) = z - A$$

on the segment $[1, \kappa]$ of the real axis defined by the *condition number* $\kappa$ of $A$:

$$\kappa = \|A\| \, \|A^{-1}\|$$

The zeros of $\psi(z)$ are precisely the eigenvalues of $A$. The precise sense of the best approximation depends again on the Lanczos type method chosen.

Since every linear operator $A$ of rank $n$ satisfies an $n$'th degree polynomial equation, namely its *characteristic equation*, Krylov vectors $A^k b$ become linearly dependent after at most $n$ iterations. This may happen even earlier if $A$ satisfies a lower degree polynomial equation. This is the case if and only if $A$ has multiple eigenvalues. For a particular right hand side $b$, Krylov vectors may become linearly dependent even earlier, if $b$ has a zero component in the direction of some of $A$'s eigenvectors. When $b$ corresponds to a first guess error contaminated by Gaussian noise, this happens with probability zero only in exact arithmetic.

There exists, by Lagrange interpolation, an $n$'th degree polynomial that vanishes at all the eigenvalues of $A$. This polynomial is the best $n$'th degree approximation to the zeros of $\psi(z)$ in any norm, and by the optimality of Lanczos methods it will emerge as the residual polynomial of the iteration after at most $n$ steps in exact arithmetic. This statement implies a proof of the finite termination of a Lanczos type iterative method applied to a linear, symmetric positive definite problem.

By the relation between the linear equation (5.22) and the quadratic optimization problem (5.23), as well as the analogous local relation between the linear equation (5.21) and the nonlinear optimization problem (5.18), the finite termination property carries over to the corresponding quadratic optimization problem and is, indeed, known as the *quadratic termination property* in literature on optimization methods (*Dennis and Schnabel (1987)*).

Apart from approximating the solution of a linear equation in a Krylov space, a Lanczos type method automatically produces approximations to the eigenvalues of the corresponding operator through the zeros of its residual polynomial at every step of the iteration. As stated above, after $n$ iterations of a Lanczos method the approximations are, in fact, exact with exact arithmetic. Before this stage, the approximations can be shown to be equivalent to the so called *Ritz values* of the iterative method. The Ritz values are constrained extrema of the Rayleigh quotient (5.23) where the domain of eigenvectors $x(t_0)$ is restricted to the $k$'th Krylov space at the $k$'th iteration step. The norm defining the goodness of an approximation is the same as the one used in the corresponding Lanczos method to define the best approximation to the solution of the linear equation $Ax = b$.

### 5.4.3. Coupling parallel variational assimilation and ensemble forecasting together

As can be judged by the previous two subsections, both variational data assimilation and ensemble forecasting entail a large number of model runs - at least a few dozen - per forecast. In 4DVAR, the number of sequential runs is twice the number of function evaluations needed in the minimization. In ensemble forecasting, there is a number of parallel runs - one per each member of the ensemble - but also a number of sequential ones that are needed to generate the initial perturbations. The number of the latter corresponds to the dimension of the Krylov space needed by the Lanczos algorithm.

All the sequential model runs have until recently been carried out strictly one after another. The goal of the study in article **(IV)** was to search for parallelization opportunities in the outer minimization loop of data assimilation. This would be in addition to parallelizing the forecast model itself and the adjoint model used in 4DVAR, which can easily be done by using the same basic parallel forecast model in assimilation as in the forecast itself. In ensemble forecasting, the different forecast members are naturally independent of one another, and hence they can be executed in parallel. However, the generation of their initial perturbations by a Lanczos method requires model evaluations in a strictly sequential fashion. The same is true for the classical quasi-Newton type minimization methods used in 4DVAR.

Before embarking on the study on variational data assimilation, a parallelization analysis of the then operational statistical data assimilation, known as Optimum Interpolation (OI), was conducted by *Isaksen (1992).* The effort involved producing a simple parallel version of the operational OI code.

It has turned out since the completion of the research in (**IV**) that parallelization outside an individual model forecast is unnecessary for variational data assimilation, for the time being, since the transposition strategy allows for efficient parallelization of even low resolution models, and because the first massively parallel computer to go operational at ECMWF turned out to have very powerful processors - in fact, faster each than the processors of the then current vector supercomputer. Therefore a relatively small number of processors will suffice to achieve acceptable throughput time for each model run. The transposition strategy was therefore modified to cover also variational data assimilation by *Isaksen and Barros (1994).*

The study of additional parallelism in the outer-most minimization loop of variational data assimilation reported in (**IV**) did, however, produce some very interesting connections between parallel variational data assimilation and ensemble forecasting. These connections may still prove useful, when the current simplifications to Kalman filtering implicit in variational data assimilation are gradually removed. An efficient alternative at that stage might also be the Fast Kalman Filter by *Lange (1990).*

The principal findings in (**IV**) point towards a close similarity between seven independent vector sets:

**1.** the most unstable perturbation directions needed for ensemble forecasting

**2.** the search direction vectors in conjugate gradient minimization methods applied in variational data assimilation

3. the search direction vectors in BFGS type limited memory Quasi-Newton type minimization methods applied in variational data assimilation

4. the best independent parallel search directions in parallel quasi-Newton type minimization methods

5. the vectors forming an optimal low-rank approximation to the Hessian matrix in quasi-Newton methods

6. the vectors in an optimal low-rank approximation to the analysis error covariance matrix necessary for Kalman filtering, and therefore for any linearized error analysis of an ensuing weather forecast, by the ensemble method or otherwise

7. the singular vectors of the tangent linear model, being by definition the eigenvectors of the composite operator consisting of the resolvents of the tangent linear model, the observation operator and the adjoint model

It appears that the span of all the above vector sets is essentially the same, and determining any of them will produce all the other sets as well. Moreover, each of the vectors in the above vector sets bears close resemblance to a counterpart vector in the other sets: vectors in each set are different approximations to the singular vectors of the model linearized around the analysis. Therefore it seems like a good idea for the future to couple parallel variational data assimilation and ensemble forecasting closely together. An operational strategy to this effect is proposed in (**IV**) and summarized further down.

The validity of the equivalence of the spans of the above vector sets is dependent on the degree of quadraticity of the optimisation problem at hand, as well as on the spectral distribution of energy in the desired atmospheric analysis and in the observation noise.

An exhaustive empirical study of the similarity suggested above has been beyond the time and resources available for the current study. However, in the article **(VI)**, a number of other studies are cited, each supportive of bridging a step in the ladder of equivalences above. For the equivalence claim missing from the other studies, article **(VI)** presents results from numerical experiments with simple variational assimilation systems that support the claim.

As was seen in the previous section, conjugate directions are crucial to an efficient parallel implementation of 4D variational assimilation. Since the eigenvectors of the linear operator $\int_{t_0}^{t_n} \mathcal{R}^*\mathcal{H}'^*Q^{-1}\mathcal{H}'\mathcal{R}dt$ are both orthogonal and conjugate, and since the most unstable of them are likely to contain most of the first guess error, the most unstable perturbation directions used in ensemble forecasting should be ideal candidates for multidimensional line search directions or parallel Hessian update directions, whichever of these parallelization strategies is chosen.

If the assimilation period is short - say, 3 to 6 hours - the most unstable perturbations are local in space. This corresponds to a domain decomposition type parallelization strategy that is known to work well for most stationary elliptic problems. If the assimilation period is longer - say, 12 to 72 hours - the most unstable perturbations become increasingly synoptic in character (*Vukicevic (1991)* and *Molteni (1992)*).

The results of the studies in *Courtier and Talagrand (1987)* and *Rabier and Courtier (1992)* provide some incentive to consider longer assimilation periods. As noted in *Courtier and Talagrand (1987)*, a longer assimilation period forces the analysis to be dynamically more coherent, because the spurious inertia-gravity waves that tend to develop into non-penalized assimilation processes have to be maintained over longer periods of time and over larger geographical distances, thereby making it likelier that they will contradict some other observations and get eliminated by the assimilation process. Because a longer assimilation period produces less noisy analyses, it would facilitate imposing a smaller penalty term on gravity modes, making the cost function more consistent dynamically. *Rabier and Courtier (1992)* conclude from their experiments that the longer the assimilation period, the better the quality of the resulting analysis. This is true in operational forecasting, too, because a larger number of observations is then available. In principle, the assimilation period could be extended backwards in time up to the limit of predictability (with exponentially decaying weighting of past observations), would the nonlinearity of the dynamics not make the convergence of the minimization difficult already for periods exceeding 48 hours.

Interestingly, it might make more sense in such a case to perform 4DVAR *backwards*, using the *final* state as the control, instead of the initial state. An iteration would start by an integration of the adjoint model backwards, stopping at a suitable point in time in the past, and starting the nonlinear model integration from there in order to produce the gradient at the final time.

Longer assimilation periods should also make successive analyses more consistent, without sacrificing the dynamical consistency of the analyzed fields, as is done in the Variational Continuous Assimilation of *Derber (1989)*. This should help in reducing forecast variability, at least to the extent that it is due to the model's activity in producing small synoptic features from analysis noise. For the utilization of parallel computing in making long assimilation periods feasible, good approximate conjugate directions of the kind needed for ensemble forecasting are vital: mere spatial data decomposition will not be sufficient. These considerations suggest the following operational scenario:

**1.** When the ensemble of forecasts is computed from linear combinations of the most unstable approximate eigenvectors of $\int_{t_0}^{t_n} \mathcal{R}^*\mathcal{H}'^* Q^{-1}\mathcal{H}'\mathcal{R}dt$, the eigenvectors are stored for the use of the parallel 4DVAR.

**2.** The parallel quasi-Newton method employed in 4DVAR uses these as approximately conjugate directions to perform either (or both) multidimensional line searches or (and) additional gradient evaluations for parallel Hessian updates, using for example the block-BFGS update of *Byrd, Schnabel and Schultz (1988-a and 1988-b).* For multidimensional searches, the current gradient is projected onto each of the orthogonal perturbation directions and one or several minimization steps may be carried out on the projected components in parallel, before combining the results into a new guess and carrying out one or more global minimization steps.

**3.** During the minimization, a block QR method is applied to the resulting block tridiagonal matrix computed from the coefficients of the descent method, resulting in a block Lanczos method to compute the eigenvalues and eigenvectors of the block tridiagonal matrix, and thereby those of $\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R} dt$. Alternatively, the increment vectors to the intermediate iterands could be used as a basis for computing the initial perturbations, since they span the same subspace. If the rank of the approximate Hessian is too small for the needs of ensemble forecasting, it could be enhanced after the minimization has converged by additional finite difference gradients incorporated via BFGS updates. By the equivalence between the inverse Hessian $(J'')^{-1}$ and the analysis error covariance matrix $\underline{\underline{P}}_{t_0}^A$:

$$\underline{\underline{P}}_{t_0}^A = (J'')^{-1}$$

derived in *Rabier and Courtier (1992)*, this results in improving the estimate of the analysis error covariance matrix.

**4.** Linear combinations of the iterand increments or singular vectors thus obtained are used to generate initial perturbations for an ensuing ensemble forecast. The size of each perturbation could be the projection of the analysis increment onto the corresponding singular vector. The ensuing ensemble of forecasts should thereby give information not only on the rate of the growth of the most unstable modes, but also on the relative contribution of each of them onto the forecast error, weighted by the projection of the forecast error over the previous 24 hour period onto each unstable mode.

The resulting operational scenario would provide full resolution 4D variational analyses over even a long assimilation period. Ensemble forecasts would obtain full resolution initial perturbations of the relevant size, computed daily from the operational model.

Another likely beneficiary of the coupling of 4DVAR to Ensemble Forecasting is Kalman Filtering. The Extended Kalman-Bucy (EKB) filter transforms the state variable by the nonlinear forward model, but it also produces an estimate of the forecast error covariance matrix $\underline{\underline{P}}_t^F$ at time $t$, which is the analysis error covariance matrix $\underline{\underline{P}}_{t_0}^A$ transformed by the tangent linear model and its adjoint

$$\underline{\underline{P}}_t^F = \mathcal{R}(t, t_0)\underline{\underline{P}}_{t_0}^A\mathcal{R}^*(t_0, t)$$

A full rank implementation of error covariance propagation is prohibitively expensive - it requires integrating the tangent linear model and adjoint models as many times as there are columns in the error covariance matrix; there are as many as there are observations. A good low rank approximation to $\underline{\underline{P}}_{t_0}^A$ is therefore the only practical possibility.

By the relation $\underline{\underline{P}}_{t_0}^A = (J'')^{-1}$, a low rank approximation to the inverse Hessian will simultaneously be an approximation to the analysis error covariance matrix. All the columns of such an approximation, be they iterand increments or the columns of the approximate inverse Hessian of the quasi-Newton method, can be transformed by the tangent linear model and its adjoint to arrive at a Krylov space estimate to the forecast error covariance matrix.

4D variational assimilation has a generic advantage over another common time dependent data assimilation scheme: continuous data assimilation. In continuous data assimilation, the model is integrated only forward, but every observation is inserted at the appropriate time, and the model state of a certain number of surrounding grid points is adjusted both spatially and temporally, in order to ensure that the model evolves smoothly and maintains all the required balance conditions. Because of the spatial nonuniformity of observation density, densely observed areas require more computational effort, creating a fairly serious load imbalance problem, both spatial and temporal, for a horizontal domain decomposition based parallelization strategy. The load imbalance is essentially due to the elaborate ’nudging’ process, which is necessary to ensure that the observations are optimally utilized, and needs a fair amount of processing at every observation. Avoiding the imbalance calls for elaborate load balancing schemes (*Hawick et al (1993)*.

In 4DVAR, all the necessary balance requirements are built into the mathematical formulation of the problem. The observations need therefore only to be processed by the observation operator and the error covariance matrix. While there is still space for some load imbalance, no ’nudging’ of the surrounding points is needed, and most observations are simply interpolated, making the issue of imbalance far less serious.

## 5.5 Parallelizing an operational limited area model

A joint project was set up between the Center for Scientific Computing, the Finnish

Meteorological Institute, IBM, Helsinki University of Technology and the University of Joensuu in 1994 to produce a parallel version of the Nordic HIRLAM model, complete enough to be used as the benchmark code in the procurement process for the new supercomputer. This task was completed - even if not with a fully optimized and debugged code - in essentially six months. The article (**V**) of this thesis summarizes the carrying out and the results of the parallelization effort of the HIRLAM model.

Since the target was a fully operational model, several broader concerns needed to be addressed, when compared with the earlier work at ECMWF. Firstly, there was the issue of input and output that needed to be handled efficiently, yet portably. Since no parallel I/O standards were available at that time, it was chosen to parallelize the GRIB file format standard for meteorological fields. A detailed technical description of this can be found in *Hietaniemi et al. (1994)*.

Secondly, the transposition strategy was chosen as the basic parallelization strategy even for such a limited area grid point model as HIRLAM. Time stepping schemes are always local in time, and therefore they pose the same data dependency problems, both in the case of spectral models and in the case of grid point models. In particular, the Helmholtz equation (5.13) is the same regardless of the discretization. It induces the same global data dependencies every time step in either case. Since also local discretizations for atmospheric models normally use a rectangular grid, it is common to use separable Fast Poisson Solvers for solving (5.13). This is the case with HIRLAM, too. Such solvers use longitudinal Fourier transforms to decompose the two-dimensional problem into a set of latitudinal tridiagonal systems for longitudinal wave numbers, that can efficiently be solved by Gaussian elimination. Such solvers are therefore virtually identical with spectral transform methods, as far as data dependencies are concerned.

The transposition strategy demonstrated its efficiency once again with HIRLAM, both in benchmark runs and in coding simplicity. The latter effect was born out by the completion of the project almost on schedule in six months, although of all the five project team members, only the author had any previous experience on weather codes, only two had previous experience on parallel computers and message passing programming, and two of the three principal coding experts had never coded in Fortran before.

The parallel code was written to be fully reproducible, in order to make a case of bitwise identical weather forecasts on any number of processors. To ease future code maintenance, all parallel coding was hidden in a separate subroutine library, and all machine dependent coding was hidden beneath a portable MPI message passing interface. Details of this implementation, as well as a system manager's guide to the code produced, can be found in *Saarinen et al. (1995)*).

## 5.6 Theoretical investigation of parallel data assimilation in aeronomy

The sixth article **(VI)** in the thesis has been written more than ten years after the earliest articles. In this last article, the main topic is no longer parallel supercomputing, but still one of operational forecasting concern. It is the problem of how best to utilize atmospheric measurements made in real time by polar orbiting satellites. The measurements targeted in this study are aeronomic, rather than meteorological. In aeronomy, the goal is to study the chemical constitution and dynamics of the stratosphere, with respect to climatic phenomena, such as the green house effect and ozone depletion.

Such studies display many similarities to operational weather forecasting. The eventual aim of the research is to build an accurate and robust data assimilation system for real time aeronomic measurements. Also, the dynamics of the stratosphere are not necessarily computed within the aeronomic model itself, but taken instead from the best available operational weather models that simulate the stratosphere in sufficient accuracy. Among these, the operational model at ECMWF - the object of most of the earlier articles in this thesis - still ranks as the first, as of 2002.

In aeronomy, the goal is primarily a long term understanding of the current and future development of stratospheric chemistry. The first assimilation task is therefore a form of nowcasting: what is a comprehensive chemical picture of the current stratosphere like? This mission is turned into long range assimilation by the sparsity of data.

The target satellites of the current study - Osiris and GOMOS - only carry out vertical soundings along the day/night borderline. Since the assimilation is initially built for a single instrument only, and since the satellites fly roughly 15 cycles per day, it takes up to three months to collect one thousand observations per latitude.

During such periods, the advective transport of chemical constituents can no longer be ignored. But since the wind fields are retrieved from meteorological forecast models, aeronomic assimilation does not have to worry about the nonlinearity of fluid dynamics, but just about the one implicit in the stratospheric chemistry itself. The strong photosensitivity and chemical reactivity of the chemical constituents involved makes it necessary to couple in also a chemical kinetic model.

The time scales of chemical kinetics are often poorly known and need to be calibrated from the assimilation data set. The time scales involved in advection are very different from the chemical ones. Because of this, data assimilation can take advantage of such partial separability of the dynamics. On the other hand, the long assimilation period needed makes the running of the forward and adjoint models for the advection slow on parallel supercomputers, even if the wind field stays unchanged in chemical calibration trials.

In the beginning, article **(VI)** summarizes a range of other studies that support the similarity of the seven vector sets discussed in subsection 5.4.3. There are one or more studies that allow most of the vector sets to be judged to be similar in span. The only missing justification is between optimal parallel search directions in parallel minimization and the other sets. As it was not possible to carry out real parallel minimization studies, this problem is addressed by theoretical and experimental analysis.

The notion of *Conjugate directions* is deemed to be essential for the parallel efficiency of parallel minimization, so that parallel search procedures will not duplicate the effort carried out by the other parallel tasks. Conjugate directions are directions that are orthogonal in the inner product induced by the symmetric positive definite operator $\int_{t_0}^{t_n} \mathcal{R}^* \mathcal{H}'^* Q^{-1} \mathcal{H}' \mathcal{R} dt$ in (5.21). As this is the operator that, close to the analysis $u$, corresponds to the forecast model and adjoint model sequence employed by 4DVAR, conjugacy means that the gradients produced by 4DVAR do not project substantially onto the spans of the other parallel search directions.

Article **(VI)** features studies with two simple one dimensional models on the separability of the chemical and dynamic assimilation on parallel computers. In particular, as the chemical kinetics need to be tried and readjusted, it makes sense to consider this part of the assimilation as a perturbation to the dynamic variational assimilation.

The first model is a nonlinear Burgers' equation model

$$\frac{\partial x}{\partial t} + x\frac{\partial x}{\partial s} - \mu \frac{\partial^2 x}{\partial s^2} = 0,$$

in which $s$ is the spatial coordinate. The domain of the equation is a one-dimensional torus that has been parameterized so as to resemble the radiosonde observation network around the 45th Northern Latitude, and the time, speed and diffusion coefficients have been matched to their atmospheric counterparts. The period of the domain $L = 28.3 \cdot 10^6 m$ and the diffusion coefficient $\mu = 10^5 \frac{m^2}{sec}$. At initial time, the true state is defined as $x(s) = 20 + 15 \sin(2\pi s/L)$. Assimilation is carried out on two subsequent days, with a 24 hour period each. Observations are exact every six hours, but the analysis initial guess is shifted in phase, frequency and amplitude from the true state, and contaminated with white noise with a smoothly spatially varying amplitude, of an order of magnitude between 5 and 25 per cent. This varying amplitude reflects the nonuniform density of the radiosonde observation network across the globe.

The system was discretized with an averaged Leap-frog scheme, following *Talagrand (1991)*, and a 4DVAR adjoint written for the model. The discretization was made with 256 spatial grid points and 192 time steps over 48 hours. This corresponds to a 256 by 128 spatial grid - or a T84 spectral model - and a 15 minute time step.

The assimilation trial consists of a two day period, over which the sinusoidal wave is transported by the Burgers' equation. The solution is observed exactly every six hours. In order to simulate the impact of high frequency observation noise and an ensuing geostrophic initialisation process, the initial guess for minimization is produced from an earlier noisy analysis through mild low pass filtering.

First, a one day minimization is carried out, and all the linearly independent search directions of the quasi-Newton or conjugate gradient method used are stored. They are each recorded at the end of day one. The first parallel assimilation steps are carried out each with an initial guess consisting of the original initial guess, to which a perturbation in the corresponding earlier search direction, extracted at the end of the first day assimilation period, is added. The magnitude of each perturbation is the same as the length of the correction step taken in that direction during the assimilation on the first day.

The gradients for all the first parallel minimization steps are computed, stored and normalized, after deducting from each of them the gradient produced by the original initial guess alone. The inner products of all these reduced gradients with all the search directions are calculated and stored in a bicovariance matrix

$$\left[< \nabla_{u^i} J(u^i), s^j >_{ij}\right].$$

Ideally, for conjugate search directions, the bicovariance matrix should be the identity matrix. The more diagonally dominant it is, the better conditioned will the recombination equation to be solved by the outer iteration be.

Here, we study the conjugacy of the search directions from the assimilation carried out 24 hours before, when used as parallel search directions in an ensuing 24 hour assimilation.

The second model is a linear advection model where simple chemical kinetics are added that relax a chemical tracer species to a discontinuous steady state over roughly 30 days.

$$\frac{\partial c}{\partial t} + v\frac{\partial c}{\partial s} - \mu\frac{\partial^2 c}{\partial s^2} = f(x),$$

where $c$ is the concentration of the tracer chemical species to be assimilated.

The source term $f(x)$ is set to a negative constant for one quarter length of the latitude, and to its positive counterpart for the subsequent quarter. The constant is chosen so that $f(x)$ linearly consumes roughly ten percent of the equilibrium concentration of $v$ during the quarter cycle, which exact amount is subsequently restored during the next quarter cycle. The equilibrium state is therefore V-shaped in one half cycle, and flat in the other half cycle. Otherwise the dimensions of the system are the same as in the Burgers' case, except that the time period is set to 30 days.

The assimilation setup is created from a constant equilibrium solution at the moment when the chemical kinetics are turned on. The goal is to retrieve this initial equilibrium from a sequence of polar orbiting narrow footprint satellite observations that observe the onset and relaxation of the kinetics over a period of 30 days, with 15 sample measurements per day, each at such a point on the latitude that it corresponds to the passing longitude of a polar orbiting satellite.

The experiments were carried out with exact measurements and subsequently with ones contaminated to various levels with Gaussian white noise. The minimization algorithm was a BFGS Quasi-Newton method. Instead of an adjoint, finite difference based Hessian approximation is used in this experiment.

The goal of this second numerical model is to use the search directions of the first assimilation trial as the parallel search direction basis for subsequent assimilation trials, where just the chemical kinetics are modified. It is carried out by computing the covariance matrix of the search directions

$$\left[< s^i, s^j >_{ij}\right].$$

The results of both numerical simulations indicate that the singular vectors retain their mutual conjugacy quite well for the entire course of the assimilation cycle. From this point of view, they appear quite good parallel search directions. On the other hand, it is less clear whether the Krylov space spanned by successive singular vectors also spans the subspace where the next analysis increment resides. In a one dimensional case, this is conceivable, but it may well be that in a three dimensional model, the dimension of the smallest sufficiently large Krylov space is too high. These simplified benchmarks do not address this latter issue.

# 6. Conclusions

The current research was motivated by a practical problem: whether massively parallel computers are the right choice for operational weather forecasting. The research carried out on either of the models studied - the global spectral model IFS and the regional grid point model HIRLAM - resulted - after a number of steps, not all of which directly involved the author - in a portable parallel version of the complete forecast suite, not just the forecast model.

Both FMI and CSC in Finland and ECMWF decided by 1995, motivated principally by benchmark results with the parallel models produced, to purchase a massively parallel, distributed memory computer with 200 processors - of a different make in each case - that became the next operational forecast engine in each country from 1996 on. The two orders were worth 40 million US dollars together, and were, to the author's knowledge, the first two worldwide by operational, non-academic scientific computing organizations for distributed memory parallel computers that are to be used as general-purpose supercomputers simultaneously running parallelized mission-critical operational codes.

Both ECMWF and the Finnish Meteorological Institute have successfully run their operational forecasting activities on massively parallel, distributed memory computers ever since, with each most recently purchasing an IBM SP parallel computer in 2002, with 1920 processors (ECMWF) and 512 processors (CSC and FMI), respectively. Many other centres have followed suite since 1996, so that as of this writing, in late 2002, an operational weather forecasting centre still using a shared memory vector supercomputer is a rare exception. Most centres are into their second or third generation massively parallel computers, featuring either vector processors or workstation clusters, but with a NUMA architecture in either case.

The transposition strategy has become universally the parallelisation strategy of choice for spectral weather models. Weather and climate forecasting centers that use the transposition strategy include ECMWF, FMI, Meteo France, Max Planck Institute in Hamburg, the U. S. national weather forecasting centre NPEC in Washington, Fleet Navy Oceanography Center in Monterrey, Japanese Meteorological Agency and the Swedish and Norvegian meteorological institutes. The German weather service DWD did, for as long as they carried out global weather forecasts. The most recent adopter is the Earth Simulator in Japan, which is the most powerful supercomputer in the world by far in 2002, and is being used exclusively for simulating the climate at very high resolution.

# 7. References

1. **Barros, S., Dent, D., Isaksen, L. and Robinson, G. (1994)**: *The IFS model - overview and parallel strategies.* Coming of age. (Hoffmann, G.-R. and Kreitz, N., eds.) 303 - 318. Proceedings of the Sixth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1994. World Scientific 1995.

2. **Barros, S. and Kauranne, T. (1990)**: *Spectral and multigrid spherical Helmholtz equation solvers on distributed memory parallel computers (1990).* Proceedings of the Fourth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1990. (Hoffmann, G.-R. and Kauranne, T., eds.). 1 - 27. ECMWF Workshop Proceedings 1992.

3. **Byrd, R. H. and Nocedal, J. (1989)**: *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization.* SIAM J. Numer. Anal. 26 (1989) 3, 727-739.

4. **Byrd, R. H., Schnabel, R. B. and Schultz, G. A. (1988-b)**: *Using parallel function evaluations to improve Hessian approximation for unconstrained optimization.* Annals of Operations Research 14 (1988), 167-193.

5. **Courant, R., Friedrichs, K.-O. and Lewy, H. (1928)**: *Über die partielle differenzengleichungen der mathematischen physik.* Math. Annalen 100 (1928), 32 - 74.

6. **Courtier, P. and Talagrand, O. (1987)**: *Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory.* Q. J. R. Meteorol. Soc. 113 (1987), 1311-1328.

7. **Dennis Jr., J. E. and Schnabel, R. B. (1987)**: *A view of unconstrained optimization.* in Handbooks in Operations Research and Management Science, Vol. 1: Optimization. (Nemhauser, G. L., Rinooy Kan, A. H. G. and Todd, M. J., editors), North-Holland.

8. **Dent, D., Hamrud, M., Haseler, J., Mozdzynski, G., Salmond, D. (2002)**: *Programming for the IBM high-performance computing facility.* ECMWF Newsletter 94 (2002), 9-10.

9. **Dent, D., Isaksen, L., Mozdzynski, G., Robinson, G., Wollenweber, F. and O'Keefe, M. (1994)**: *The IFS model - performance measurements.* Coming of age. (Hoffmann, G.-R. and Kreitz, N., eds.) 352 - 369. Proceedings of the Sixth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1994. World Scientific 1995.

10. **Derber, J. C. (1989)**: *A variational continuous assimilation technique.* Mon. Weather Rev. 117 (1989), 2437-2446.

11. **Fisher, M. (1998)**: *Development of a simplified Kalman filter.* Technical Memorandum No. 260, European Centre for Medium-range Weather Forecasts, Reading 1998.

12. **Fisher, M. and Courtier, P. (1995)**: *Estimating the covariance matrices of analysis and forecast error in variational data assimilation.* Technical Memorandum No. 220, European Centre for Medium-range Weather Forecasts, Reading 1995.

13. **Foster, I. and Worley, P. (1994)**: *Parallel algorithms for the spectral transform method.* Technical Report ORNL/TM-12507, Oak Ridge National Laboratory, Oak Ridge 1994.

14. **Gärtel, U., Joppich, W., Mierendorff, H. and Schüller, A. (1994-1)**: *Performance of the IFS on parallel machines - measurements and predictions.* Coming of age. (Hoffmann, G.-R. and Kreitz, N., eds.) 319 - 338. Proceedings of the Sixth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1994. World Scientific 1995.

15. **Gärtel, U., Joppich, W. and Schüller, A. (1994-2)**: *Portable parallelization of the ECMWF's weather forecast program - technical documentation and results.* Arbeitspapiere der GMD 820, GMD Sankt Augustin 1994.

16. **Golub, G. H. and Van Loan, C. F. (1989)**: *Matrix computations. (second edition)* The Johns Hopkins University Press 1989.

17. **Hawick, K. (1994)**: *Parallelization of the Unified Model data assimilation scheme.* Parallel supercomputing in atmospheric science. (Hoffmann, G.-R. and Kauranne, T., eds.) 188 - 203. Proceedings of the Fifth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1992. World Scientific 1993.

18. **Hietaniemi, J., Oinonen, J., Kauranne, T., Saarinen, S. and Serimaa, O. (1994)**: *Parallel I/O on meteorological GRIB files.* Coming of age. (Hoffmann, G.-R. and Kreitz, N., eds.) 91 - 109. Proceedings of the Sixth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1994. World Scientific 1995.

19. **Hockney, R. and Curington, I. (1989)**: $f_{1/2}$*: a parameter to characterize memory and communication bottlenecks.* Parallel Computing 10 (1989), 277 - 286.

20. **Hoffmann, G.-R. and Kreitz, N., eds. (1995)** *Coming of Age.* Proceedings of the Sixth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1994. World Scientific 1995.

21. **Hoffmann, G.-R. and Kreitz, N., eds. (1997)** *Making its Mark.* Proceedings of the Seventh Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1996. World Scientific 1997.

22. **Isaksen, L. (1992)**: *Parallelizing the ECMWF optimum interpolation analysis.* Parallel supercomputing in atmospheric science (Hoffmann, G.-R. and Kauranne, T., eds.) 240 - 249. Proceedings of the Fifth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1992. World Scientific 1993.

23. **Isaksen, L. and Barros, S. (1994)**: *IFS 4D-variational analysis - overview and parallel strategy.* Coming of age. (Hoffmann, G.-R. and Kreitz, N., eds.) 337 - 351. Proceedings of the Sixth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1994. World Scientific 1995.

24. **Kauranne, T. (1988-1)**: *Convergence rate of the conjugate gradient method applied to linear elliptic equations.* Proceedings of the First Copper Mountain Conference on Iterative Methods, Copper Mountain April 1990. University of Colorado in Denver, Computational Mathematics Group, Denver 1990. Also

published as a Master's thesis at the Department of Computer Science, University of Helsinki, 1988.

25. **Kauranne, T. (1988-2)**: *An Introduction to parallel computing in meteorology.* The Dawn of Massively Parallel Processing in Meteorology: Proceedings of the Third Workshop on the Use of Parallel Processors in Meteorology, ECMWF December 1988. (Hoffmann, G.-R. and Maretis, D., eds.) 3 - 20. Springer-Verlag, Berlin 1990.

26. **Kauranne, T. (1988-3)**: *Asymptotic parallelism of weather models.* The Dawn of Massively Parallel Processing in Meteorology: Proceedings of the Third Workshop on the Use of Parallel Processors in Meteorology, ECMWF December 1988. (Hoffmann, G.-R. and Maretis, D., eds.) 303 - 314. Springer-Verlag, Berlin 1990.

27. **Kauranne, T. (1990)**: *Preliminary estimates on the performance of a spectral weather model on the Suprenum parallel computer.* Final Report of the GENESIS project to the Esprit II programme of the European Commission, Brussels 1990.

28. **Kauranne, T. (1991)**: *Global atmospheric models on massively parallel computers.* Licenciate Thesis, University of Jyväskylä; 1991.

29. **Kauranne, T. and Hoffmann, G.-R. (1990)**: *Parallel processing: a view from ECMWF.* Proceedings of the Fourth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1990. (Hoffmann, G.-R. and Kauranne, T., eds.). 148 - 169. ECMWF Workshop Proceedings 1992.

30. **Lacarra, J.-F. and Talagrand, O. (1988)**: *Short-range evolution of small perturbations in a barotropic model.* Tellus 40A (1988), 81-95.

31. **Lange, A. (1990)**: *Real-time optimum calibration of large sensor systems by Kalman filtering.* Proceedings of the IEEE 1990 PLANS - Position, Location And Navigation Symposium. Las Vegas, March 1990.

32. **Le Dimet, F.-X. and Navon, I. M. (1996)**: *Second order information in data assimilation: a review.* A manuscript. 1996.

33. **Le Dimet, F.-X. and Talagrand, O. (1986)**: *Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects.* Tellus 33A (1986) 97 - 110.

34. **Liu, D. C. and Nocedal, J. (1989)**: *On the limited memory BFGS method for large scale optimization.* Mathematical Programming 45 (1989) 503 - 528.

35. **Molteni, F. (1992)**: *Personal communication.*

36. **Molteni, F., Mureau, R. and Palmer, T. N. (1991)**: *Atmospheric instability and ensemble weather prediction.* ECMWF Technical Memorandum No. 177.

37. **Molteni, F. and Palmer, T. N. (1991)**: *Predictability and finite-time instability of the northern winter circulation.* Proceedings of the ECMWF Workshop on New developments in predictability, November 1991, 101-142.

38. **Mureau, R., Molteni, F. and Palmer, T. N. (1991)**: *Ensemble predictions using dynamically conditioned perturbations.* Proceedings of the ECMWF Workshop on New developments in predictability, November 1991, 143-174. ECMWF Workshop Proceedings, ECMWF 1991.

39. **Nazareth, L. (1979)**: *A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms.* SIAM J. Numer. Anal. 16 (1979) 5, 794-800.

40. **Nazareth, L. (1986)**: *Conjugate gradient methods less dependent on conjugacy.* SIAM Review 28 (1986) 4, 501-511.

41. **Rabier, F. and Courtier, P. (1992)**: *Four-dimensional assimilation in presence of baroclinic instability.* Q. J. R. Meteorol. Soc. 118 (1992), 649-672.

42. **Saarinen, S., Oinonen, J., Hietaniemi J., Kauranne, T. and Serimaa, O. (1995)**: *Parallel HIRLAM user's guide.* Technical Report, Centre for Scientific Computing, Espoo 1995.

43. **Shingu, S., Fuchigami, H., Yamada, M. (2002)**: *A 26.8 Tflops spectral atmospheric model on the Earth Simulator.* To appear in: Realizations of Teracomputing. (Zwiefelhofer, W. and Kreitz, N., eds.) Proceedings of the Tenth

Workshop on the Use of High Performance Computing in Meteorology, ECMWF November 2002. World Scientific 2003.

44. **Simmons, A. (1994)**: *High-performance computing reuirements for medium-range weather forecasting.* Coming of age. (Hoffmann, G.-R. and Kreitz, N., eds.) 1 - 11. Proceedings of the Sixth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1994. World Scientific 1995.

45. **Talagrand, O. (1991)**: *Adjoint models.* Proceedings of the ECMWF Seminar on Numerical methods in atmospheric models, ECMWF September 1991, Volume II, 73 - 91. ECMWF Seminar Proceedings, ECMWF 1991.

46. **Tremolet, Y. and Le Dimet, F.-X. (1996)**: *Parallel algorithms for variational data assimilation and coupling models.* Parallel Computing 22 (1996), 657 - 674.

47. **van der Sluis, A. and van der Vorst, H. A. (1986)**: *The rate of convergence of conjugate gradients.* Numerische Mathematik 48 (1986) 543 - 560.

48. **Vukicevic, T. (1991)**: *Nonlinear and linear evolution of initial forecast errors.* Mon. Weather Rev. 119 (1991), 1602-1611.

49. **Worley, P., Foster, I. and Toonen, B. (1994)**: *Algorithm comparison and benchmarking using a parallel spectral transform shallow water model.* Coming of age. (Hoffmann, G.-R. and Kreitz, N., eds.) 277 - 289. Proceedings of the Sixth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1994. World Scientific 1995.

50. **Zhi, W., Navon, I. M. and Zou, X. (1993)**: *The adjoint truncated Newton algorithm for large-scale unconstrained optimization.* Manuscript accepted for publication in Optimization Methods and Software (1993).

51. **Zou, X, Navon, I. M., Berger, M., Phua, K. H., Schlick, T. and Le Dimet, F.-X. (1993)**: *Numerical experience with limited memory Quasi-Newton and Truncated Newton methods.* SIAM J. Optimization 3 (1993) 3, 582 - 608.

52. **Zwiefelhofer, W. and Kreitz, N., eds. (1999)** *Towards Teracomputing.* Proceedings of the Eighth Workshop on the Use of Parallel Processors in Meteorology, ECMWF November 1998. World Scientific 1999.

53. **Zwiefelhofer, W. and Kreitz, N., eds. (2001)** *Developments in Teracomputing.* Proceedings of the Ninth Workshop on the Use of High Performance Computing in Meteorology, ECMWF November 2000. World Scientific 2001.